AUTUMN 1984

An Argus Specialist Publication

Personal SOFTWARE

£2.10

Great programs for the Oric and Atmos

Mazes, snakes and aliens – can you handle the action?

Understand your micro and what it can do

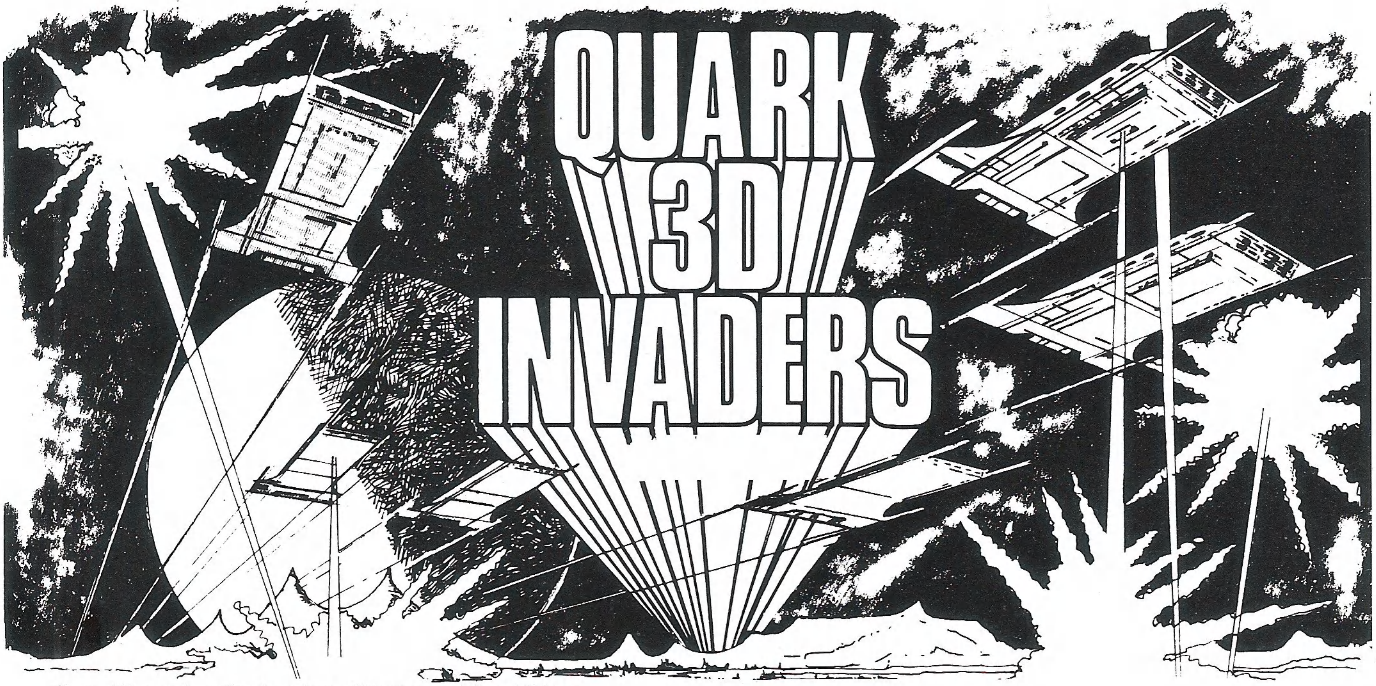
Useful routines and programs to help you along the way

Racey, pacey games to get you going

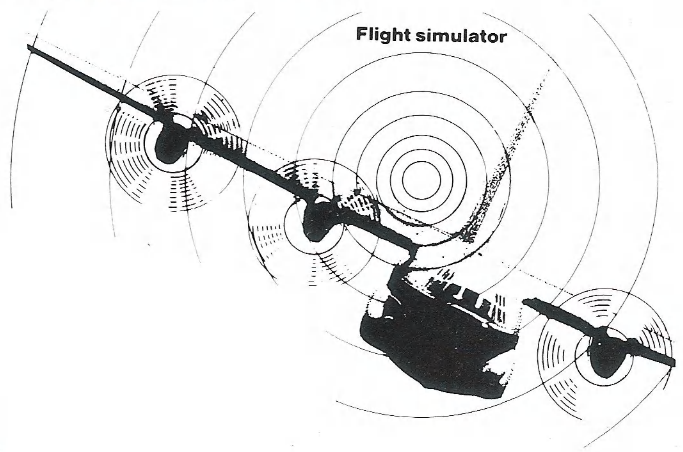
ORIC
ORIC ATMOS 48K

Full user reports inside.

ORIC & ATMOS 48K Cassettes



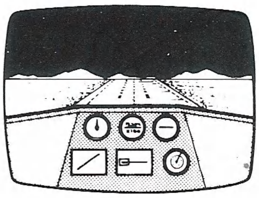
Super 3D arcade action Invaders will test your skill and 3D co-ordination to their limits. 8 rippling colour Invader screens of increasing difficulty with keyboard controlled 3D laser movement and fire sequence. (Brain required) **£6.95**



Oricle *(Smart Oric)*
Hours of family fun. Oric gets smarter the longer you play, eventually you will have no secrets. Be careful what you say. **£6.95** *Also in French)*

Mystery Tower
Superb musical arrangements. 47 different rooms. 3D and colour make this adventure game most enjoyable. You will certainly get lost, or forget where you're going. **£6.95** *(Also in French)*

Flight Simulator
With airspeed, radar, ground/vertical flight path indicators, artificial horizon and altimeter with continuous flight monitor on engine condition, stall speed etc., full colour, Hi-res graphics, 3D and sound. **£6.95**



Space Crystal
Space trading adventure game. Fascinating 3D space travel on your voyage in search of the illudium crystal, Black holes, meteorite storms, Space pirates, etc. Full colour and sound. **£6.95.** *Also available for Dragon 32, Spectrum and Commodore 64*

QUARK DATA P.O. BOX 61, Swindon, Wilts. Tel. (0793) 40661.
Please supply. . . . .
. . . .
Prices incl. p & p UK only.
Cheque for total amount enclosed . . . .
Name . . . .
Address . . . .
. . . .

Access

AUTUMN 1984
An Argus Specialist Publication
Personal
SOFTWARE
Great programs for the Oric and Atmos
Mazes, snakes and aliens - can you handle the action!
Understand your micro and what it can do
Useful routines and programs to help you along the way
Full user reports inside.

Volume 3
No 2
Autumn 1984

Editor: Wendy J Palmer
Sales Executive:
Penny Scoular
Advertisement Manager:
Barry Bingham
Advertisement Copy Control:
Sue Couchman
Chief Executive: T J Connell
Origination and design by MM Design & Print

Personal Software is a quarterly magazine appearing on the third Friday in May, August, November and February.

Distribution by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, London EC2A 4JS Tel: 01-247 8233. Printing by: Alabaster Passmore & Sons Ltd, Tovil, Maidstone, Kent.

The contents of this publication including all articles, designs, plans, drawings and programs and all copyright and other intellectual property rights therein belong to Argus Specialist Publications Limited. All rights conferred by the Law of Copyright and other intellectual property rights and by virtue of international copyright conventions are specifically reserved to Argus Specialist Publications Limited and any reproduction requires the prior written consent of the Company. © 1984 Argus Specialist Publications Limited.

Subscription Rates : UK £8.40 for four issues including postage. Airmail and other rates upon application to Personal Software Subscriptions, Infonet Ltd, Times House, 179 The Marlowes, Hemel Hempstead, Herts HP1 1BB.

CONTENTS

Editorial & Advertisement Office
No 1 Golden Square, London W1R 3AB
Telephone: 01-437 0626 Telex: 8811896

Astro Attack 5
Those aliens have started attacking again.

Orcman 8
This is what somebody removed from today's maze game characters.

School of Death 10
You have to find the hidden 'treasure' in your brother's school.

Monster Hunt 16
A great maze/adventure game where you hunt the monsters.

Escape from Alpha Centuri 19
Can you take pity on and help an Alpha Centuri inhabitant escape.

Wumpus Hunt 22
Getting into the same square as a Wumpus may sound easy, but . . .

Oric Options 28
We introduce you to some of the less well-known facilities of the Oric.

Oric Demo 35
So you just bought an Oric? Type this neat little program in.

Snake 36
Those little devils are out and growing by the minute.

Alien Dropout 39
Your mission is to defend your planet.

3D Maze 42
It's amazing just how confusing this sort of game can be!

Hiresalt City 44
See how good you are at driving your runner around the city.

Chord Chorus 46
This is for those of you who need to learn about musical chords.

Twister 49
Like playing cards? Why not try your hand at playing this version of Pontoon.

Machine Code 52
Whether you're already writing machine code or just learning, this disassembler will make life easier.

Forest Trek 56
Playing this game may be a case of not seeing the wood for the trees!

Zap 'em 58
Can you save your country from the fleet of enemy ships?

Help An Alien 60
Instead of zapping aliens, why not help this one back to his spaceship.

Reversi 62
Turn your life upside down with this game of stategy.

Toolkit 66
Machine code causes untold problems for programmers.

Fruit Machine 69
Here is a chance to play one of those addictive games.

Santa's Problem 72
Can you help Santa.

Missile Attack 74
You are in charge of the combat computer engaged in saving cities.

Oric Action 78
Three great games to exercise your wits.

Pyramid and Electric Eel 82
Pyramids and eels don't normally go together but. . .

Drawing The Line 85
Make your Oric quicker on the draw with this nifty little program.

Alien Scouts 87
A wave of alien scouts are about to destroy your city.

Oric Review 90
We take a look at what the Oric has to offer.

Atmos Review 94
How much does the Atmos differ from the Oric?

Escape from Danger Island 96
Earn your fare and you can pay to get off Danger Island.

Credits: Our grateful thanks to the following individuals for their assistance in the production of this issue. Mark, Enzo, Gary, Ed, Ian, William, Paul, and others almost too numerous to mention.

Wendy J Palmer

PREFACE

Well, the summer is almost gone and the nights will soon start drawing in and what will computer users do then, poor things? If they have any sense at all, they will plug in their trusty Orics and Atmos micros, get out this copy of Personal Software and get set for some fun, frolic and instruction. This issue is loaded with terrific programs for Oric/Atmos owners, collected from our sister magazines, Games Computing, Personal Computing Today and Home Computing Weekly.

GREAT STUFF

We have packed this magazine with lots of games to test your abilities — save yourself from attack by those inevitable aliens, find your way out of mazes and generally play yourself to exhaustion! There is also some more serious material for those of you who would like to learn more about how to use your micro to the fullest.

You will find that most of the programs give you tips on how to convert the games for other micros. This is obviously of help if you happen to own a different micro and just enjoy the challenge of converting programs, but you will also find the notes helpful in understanding how the program makes the micro do what it does!

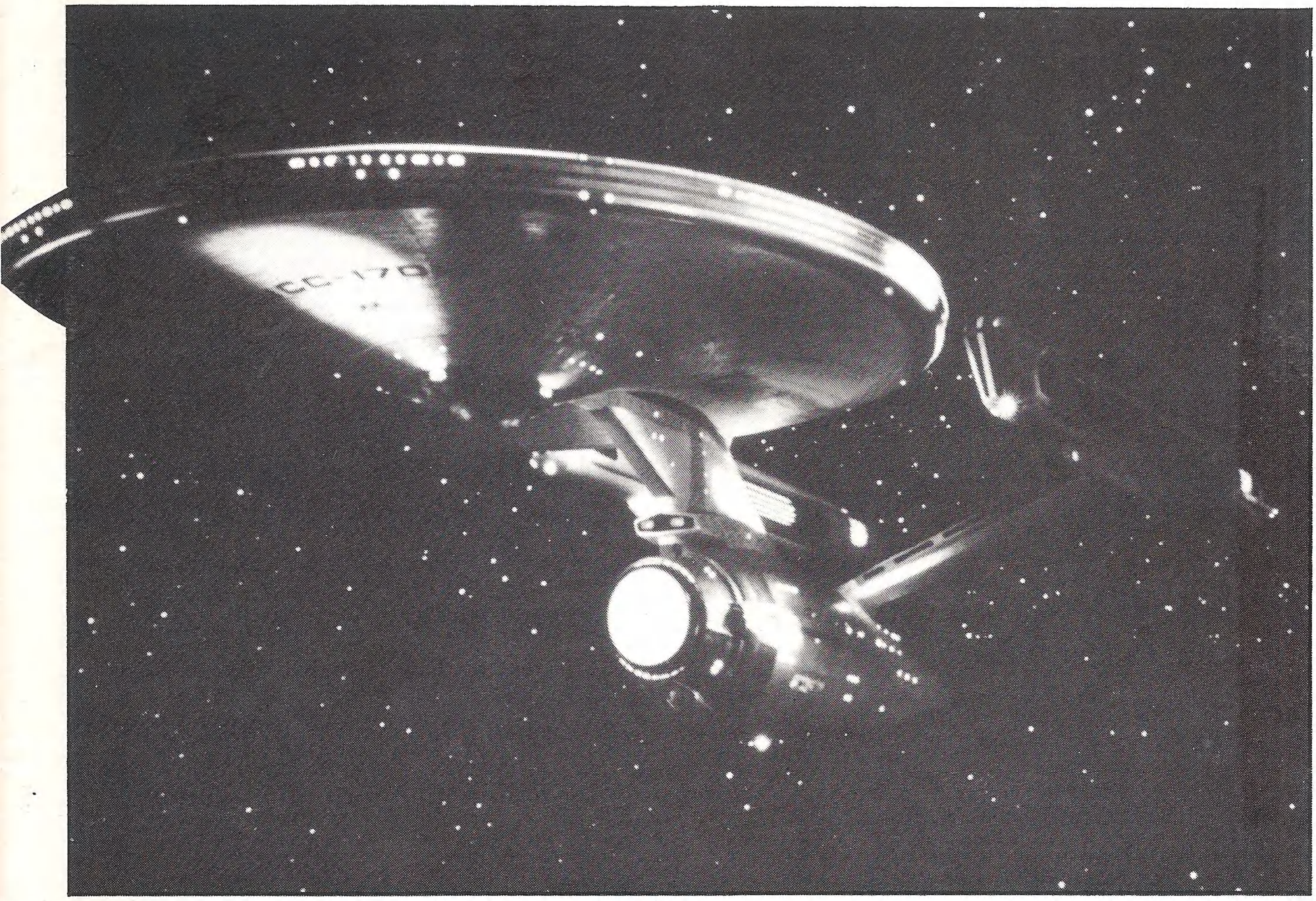
If you own an Atmos you may experience some difficulty getting some of the programs to run — again the conversion and explanatory notes in the articles should help you make the changes.

TAKE CARE . . .

As I said earlier, all of the programs and articles in this Personal Software have been published before — we are reasonably confident that there are no errors in any of the listings. If you are having problems getting a program to run, please re-check your listing — a single comma missed out in a vital place can mean death to an otherwise 'perfect' program! Why not enlist the help of a friend to check your input against the published listing — we all know that looking for that missing comma can be a lot more mind-boggling than looking for the proverbial needle in the haystack! If all else has failed, you may want to write to us here at the office. If you do, please enclose an SAE and state very clearly exactly what the problem is. We will endeavour to answer all enquiries, but we are unable to do anything over the phone, so please do not call the office — write.

ASTRO ATTACK

Those aliens have started attacking again — can you save your ship?



Astro Attack combines all the Oric colours with a relatively fast and addictive space game.

When the program is run, user defined graphics are set up and the instructions given to you. The program asks you to select a skill level which determines your score and speed of the game. My personal best score was 2532 at the expert's level. You are awarded a bonus of 600 points after 1200 points have been scored.

Once the instructions are over, the screen is set up with the game. On the top right hand corner of the screen is your score, with the high score, fuel supply and lives left recorded on the top left of the screen.

Now for the action. Your ship appears half way down the screen on the left hand side of it, with the enemy putting in an appearance to the right of the screen. Control your own ship using the UP and DOWN cursor keys respectively, the RIGHT cursor key to fire your weapons.

If you choose expert level for skill you lose 10 points if the enemy ship gets past you. Expert level is also faster.

When your fuel is exhausted by gradually going down during the game, you lose a life and start a new game. You also lose lives by crashing into the moving landscape. When all three lives are lost the game ends and you are asked if you want to play again.

When the game starts the enemy ship moves across the screen, weaving up and down as it flies. You just fire your laser gun and shoot down the ship. The game works on a score board. For instance if you try your hand at the expert level you lose 10 points if you allow an enemy ship to get past your own ship. This game runs on a 48K Oric. ▶

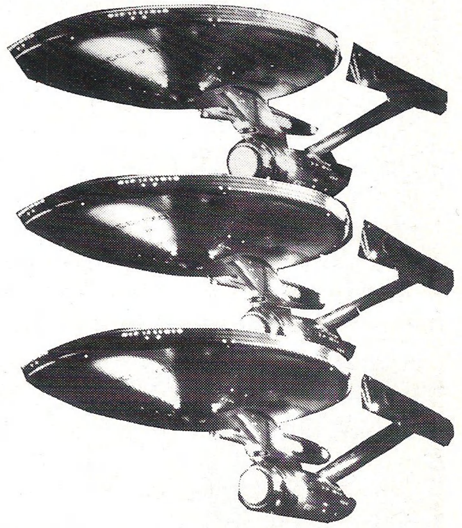
CONVERSION CLUES

POKE 618,10	Turns the cursor and keyboard click off.
POKE 48035,0	Should be omitted if programmed for other computers.
PLOT	Should be replaced with PRINT AT.
KEY$	Should be replaced by INKEY$.

Colour codes are:

0, BLACK.
1, RED.
2, GREEN.
3, YELLOW.
4, BLUE.
5, MAGENTA.
6, CYAN.
7, WHITE.

ZAP and EXPLODE	Should be replaced by your own machine's equivalent commands.



If you only have a 16K Oric you can use the program by changing the POKE addresses in which case your best bet is to have a peek at p145 of your Oric user manual.

RUNdown

Lines	Action
10-60	Initialization.
70-270	Set up screen.
300-370	Move your ship.
380-420	Move enemy ship.
430-470	Move landscape.
480	Plots score.
510	Deletes enemy ship.
535	Checks to see if enemy ship has passed you.
540	Checks fuel condition.
560-570	No fuel left.
580	Checks amount of lives left.
600-620	Bonus 600.
640-740	End of game.
760-970	Instructions.
990-1060	User-defined graphics.

VARIABLES USED

LL = Lives left.
SC = Score.
HI = High score.
F = Fuel.
Y = Position of your ship.
T,X = Position of enemy ship.
A$,B$,C$ = Landscape.
SS$ = Enemy craft.
CR$ = Your craft.
K$ = Keyboard inputs.
SK = Skill level.
A,H,I = General purpose variables.

```
0 CLS
5 REM ** ASTRO ATTACK BY G.BAILEY **
10 POKE618,10:POKE48035,0
20 PAPER0:INK6
30 GOSUB990:HI=0
40 LL=3:SC=0:A=0:X=0
50 GOSUB760
60 CLS
⊠70 REM ********** SCREEN **********
⊠80 INK5:POKE618,10:POKE48035,0
90 PRINTCHR$(12)
100 A$="♀♀♀♀♀♀♀⊠♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀
♀♀♀♀"
110 B$="♀    ♀♀♀♀♀   ♀♀ ♀♀     ♀♀♀♀♀  ♀ 
 ♀ "
120 C$="     ♀♀ ♀      ♀       ♀♀  ♀  ♀ 
 ♀ "
130 PLOT0,26,CHR$(9)+CHR$(4)+A$
140 PLOT0,25,CHR$(9)+CHR$(6)+B$
150 PLOT0,24,CHR$(9)+CHR$(6)+C$
160 PLOT2,1,"Hi-score "+STR$(HI)+CHR$(7)
:PLOT25,1,"Score "+STR$(SC)

170 PLOT 0,1,7:SS$=CHR$(9)+"%& "
180 CR$=CHR$(9)+CHR$(2)+CHR$(34)+"#$"
190 PLOT0,10,CR$
200 Y=10:X=16
210 PLOT5,3,17:PLOT12,3,19:PLOT30,3,16
220 PLOT0,3,"Fuel":F=30
2[illegible] RT=6TO23:PLOT5,T,5:NEXT
[illegible] IF LL=1THEN260
250 A=20:FORI=1TOLL-1:A=A+5:PLOTA,5,CR$:
NEXT
260 FORT=11TO12:PLOT15,T,CHR$(1)+CHR$(10
)+"Ready":NEXT
270 WAIT 200:FORT=11TO12:PLOT15,T,"
       ":NEXT:REM 7 SPACES
280 REM ********** PLAY **********
290 FORT=34TO5STEP-1
300 PLOT0,Y,CR$:P⊠OT2,Y+1,"   ":PLOT2,Y-
1,"   ":REM 3 SPACES
310 K$=KEY$:IFK$=""THENK$=L$
320 IF K$=CHR$(9)THENZAP:IFX=YTHENSHOOT
 :SC=SC+INT(T):GOTO510
330 IF K$=CHR$(11)THENY=Y-1:L$=K$
340 IF K$=CHR$(10)THENY=Y+1:L$=K$
350 IF Y=24THENEXPLODE:LL=LL-1:IFLL=0THE
N640
360 IFY=24 THEN WAIT 300:GOTO70
370 IF ~<7 THENY=6
380 A=RND(1)
390 IF X<8ANDA<.5THENA=.6
4~0 IF X>20 ANDA>.5THENA=.4
410 IF A>.5THENX=X+1:PLOTT,X,SS$:PLOTT+2
,X-1,"   "
420 IF A<.5THENX=X-1:PLOTT,X,SS$:PLOTT+2
,X+1,"   "
430 L2$=LEFT$(B$,1):R2$=RIGHT$(B$,35)
440 L3$=LEFT$(C$,1):R3$=RIGHT$(C$,35)
450 B$=R2$+L2$:C$=R3$+L3$
460 PLOT0,25,CHR$(9)+CHR$(6)+B$
470 PLOT0,24,CHR$(9)+CHR$(6)+C$
480 PLOT25,1,"Score "+STR$(SC)+"  "
490 IF SC>1200 AND SC<1700 THEN GOSUB600
495 IF SK=0THENWAIT10
500 NEXT
510 PLOTT,X,"     ":REM 5 SPACES
520 X=INT(RND(1)*10+5):F=F-1
530 PLOT F,3,16
535 IFT=4ANDSK=1THENSC=SC-10
540 IF F=5THEN 560
545 PLOT25,1,"Score "+STR$(SC)+"   "
550 GOTO280
560 FORT=11TO12:PLOT12,T,CHR$(8)+CHR$(10
)+CHR$(6)+"Fuel Exhausted":
NEXT
570 WAIT300:LL=LL-1
580 IF LL=0THENGOTO640
590 GOTO 70
600 FORH=11TO12:PLOT12,H,CHR$(8)+CHR$(6)
+CHR$(10)+"BONUS 600":NEXT
610 FORH=SCTOSC+600:PLOT31,1,STR$(H):NEXT
620 SC=SC+600:FORH=11TO12:PLOT12,H,"
       ":~EXT:REM 12 SPACE
S
630 RETURN
640 REM ******** END OF GAME *******
⊠50 FORT=15TO16:PLOT13,T,CHR$(8)+CHR$(10
)+CHR$(5)+"GAME OVER":NEXT
660 WAIT 300:CLS
670 INK3
680 PRINT:PRINT,,," High Score ";HI
```

```
690 PRINT:PRINT:PRINT"Your score was ";S
C
700 IF SC>HITHENHI=SC
710 PRINT:PRINT:PRINT:PRINT:PRINT,,,"Aga
in (Y/N) ?"
720 REPEAT:K$=KEY$:UNTILK$="Y"OR K$="N"
730 IF K$="Y"THENGOTO40
740 CLEAR
750 END
760 REM ******* INSTRUCTIONS *******
770 PRINTCHR$(12):INK6
780 PRINTCHR$(4);CHR$(27)"J             A
STRO ATTACK":PRINTCHR$(4)
790 PLOT0,1,3:PLOT0,2,1
800 PRINT:PRINT:PRINT" Your mission is t
o destroy the plague"
810 PRINT"carrying space-craft from the
planet "
820 PRINT:PRINT"of Dione."
830 PRINT:PRINT" With your laser zap dow
n the weaving"
840 PRINT:PRINT"craft by using the right
 cursor key."
850 PRINT:PRINT" Position your ship usin
g the up and"
860 PRINT:PRINT"down cursor keys."
870 PRINT:PRINT:PRINT" You have 3 lives
and lose 1 every "
880 PRINT:PRINT"time you exhau⊠t your fu
el."
890 PRINT:PRINT:PRINT,,"Press <SPACE> to
 continue"
900 REPEAT:UNTILKEY$=" "
910 CLS
920 PRINT:PRINT:PRINT" Don't crash into
the moving landscape"
930 PRINT"unless you're a kamikaze pilot
."
940 FORT=15TO16:PLOT~1,T,CHR$(10)+CHR$(2
)+"GOOD LUCK !":NEXT
950 PRINT:PRINT:PRINT"Press A for amateu
r or E for expert."
955 REPEAT:K$=KEY$:UNTILK$="A"ORK$="E"
960 IF K$="E"THENSK=1ELSESK=0
965 PRINT:PRINT:PRINT,,,"Press <SPACE> t
o play":REPEAT:UNTILKEY$="
"
970 RETURN
980 END
990 FORT=47104+(8*33)TOT+47:READD:POKET,
D:NEXT
1000 DATA 63,63,63,63,63,63,63,63
1010 DATA 63,63,0,7,7,0,63,63
1020 DATA 32,63,63,63,63,63,63,32
1030 DATA 0,60,3,63,63,62,60,0
1040 DATA0,15,31,50,50,31,15,0
1050 DATA 0,60,62,19,19,62,60,0
1060 RETURN
```

Alasdair Jarvie

ORCMAN

This is somewhat removed from today's mazes — just what are the ghosts up to?

Orcman is set far into the future, long after all today's maze-game characters have withered away.

The ghosts are fed up with chasing Orcman and only have the occasional snack.

VARIABLES

Time	Time (what else?).
Q	Score.
PR	Dots.
A,B,C,D	Positions of Orcman.
E,F,G,H	Ghost 1 positions.
I,J,K,L	Ghost 2 positions.

Orcman's predecessors have eaten all the power bills, so he can't gobble up the ghosts.

He has to eat the dots without bumping into a ghost; in fact you only need to eat 126 dots.

The ghosts have also changed. Now they are workaholics and lay dots. But they don't like interruptions. If you interrupt them it's the end for you.

If you make it and finish a screen the game sends you to another screen.

To add some more tension there is a time limit.

Use the cursor keys to guide Orcman. And good luck . . .

HOW IT WORKS

10	Ensures text mode.
20	If on, turns off cursor and key click.
30-40	Call subroutines for redefinition and instructions.
50-60	Clear screen, change colour.
70-320	Plot screen.
330-360	Define variables.
370-420	Plot colours.
430-480	Get input and respond.
490-590	Randomise ghosts' movements.
600-870	Orcman moves left, plot time and score, random movement of ghosts.
880-1150,	
1160-1430,	
1440-1700	Same for right, down, up.
1710-1800,	
1810-1890,	
1900-1980,	
1990-2070	Ghost 1 moves left, right, down, up.
2080-2160,	
2170-2250,	
2260-2340,	
2350-2430	Ghost 2 moves left, right, down, up.
2450-2650	Redefine characters.
2660-2890	Instructions.
2900-3080	Grand finale.

HINTS ON CONVERSION

Plot	Plots characters on screen.
SCRN	See what is on screen.
CLS	Clear screen.
PAPER, INK	Change colours.
MUSIC	Plays music.

```
10 TEXT
20 POKE 618,10
30 GOSUB 2450
40 GOSUB 2660
50 INK3:PAPER0
60 CLS
70 PLOT 1,1,"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
80 PLOT 1,2,"abcbcbcbcbcbcbcbcfgfgbcbcbcbcbcbcbcbca"
90 PLOT 1,3,"adedededededededehihidededededededededea"
100 PLOT 1,4,"abcfgfgfgbcfgfgbcbcbcbcfgfgbcfgfgfgbca"
110 PLOT 1,5,"adehihihidehihidedededehihidehihihidea"
120 PLOT 1,6,"abcfgbcbcbcbcfgbcfgfgbcfgbcbcbcbcfgbca"
130 PLOT 1,7,"adehidedededehidehihidehidededededehidea"
140 PLOT 1,8,"abcfgbcfgfgbcfgbcfgfgbcfgbcfgfgbcfgbca"
150 PLOT 1,9,"adehidehihidehidehihidehidehihidehidea"
160 PLOT1,10,"abcbcbcbcbcbcbcbcfgfgbcbcbcbcbcbcbcbca"
170 PLOT1,11,"adedededededededehihidededededededededea"
180 PLOT1,12,"abcfgfgfgfgfgfgbcbcbcbcfgfgfgfgfgfgbca"
190 PLOT1,13,"adehihihihihihidededededehihihihihihidea"
200 PLOT1,14,"abcbcbcbcbcbcfgbcfgfgbcfgbcbcbcbcbcbca"
210 PLOT1,15,"adedededededehidehihidehidedededededea"
220 PLOT1,16,"abcfgbcfgfgbcbcbcfgfgbcbcbcfgfgbcfgbca"
230 PLOT1,17,"adehidehihidededehihidededehihidehidea"
240 PLOT1,18,"abcfgbcfgfgbcfgbcfgfgbcfgbcfgfgbcfgbca"
250 PLOT1,19,"adehidehihidehidehihidehidehihidehidea"
260 PLOT1,20,"abcfgbcbcbcbcfgbcfgfgbcfgbcbcbcbcfgbca"
270 PLOT1,21,"adehidedededehidehihidehidededededehidea"
280 PLOT1,22,"abcfgfgfgbcfgfgbcbcbcbcfgfgbcfgfgfgbca"
290 PLOT1,23,"adehihihidehihidedededehihidehihihidea"
300 PLOT1,24,"abcbcbcbcbcbcbcbcfgfgbcbcbcbcbcbcbcbca"
310 PLOT1,25,"adedededededededehihidededededededededea"
320 PLOT1,26,"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
330 REM **** VARIABLES *****
340 Q$="jk":W$="lm":E$="no":R$="pq"
350 A=20:B=12:C=20:D=13:E=10:F=6:G=10:H=7:TIME=230
360 I=28:J=10:K=28:L=11
370 REM *** MAIN GAME ***
380 FOR Y=0TO25STEP2
```

```
390 COLOUR=INT(RND(1)*6)+1
400 PLOT 0,Y,COLOUR
410 PLOT 0,Y+1,COLOUR
420 NEXTY
430 A$=KEY$
440 PLOT A,B,Q$:PLOT C,D,W$:PLOT E,F,E$:PLOT G,H,R$:PLOT I,J,E$:PLO
TK,L,R$
450 IF A$=CHR$(8)THENGOSUB600
460 IF A$=CHR$(9)THENGOSUB880
470 IF A$=CHR$(10)THENGOSUB1160
480 IF A$=CHR$(11)THENGOSUB1440
490 GH1=INT(RND(1)*4)
500 IFGH1=0THEN GOSUB 1720
510 IFGH1=1THEN GOSUB 1810
520 IFGH1=2THEN GOSUB 1900
530 IFGH1=3THEN GOSUB 1990
540 GH2=INT(RND(1)*4)
550 IFGH2=0THEN GOSUB 2080
560 IF GH2=1THEN GOSUB 2170
570 IFGH2=2THEN GOSUB 2260
580 IFGH2=3THEN GOSUB 2350
590 GOTO 430
600 REM **** LEFT *****
610 IF SCRN(A-1,B )=ASC("g")THENRETURN
620 IF SCRN(A-1,B)=ASC("a")THENRETURN
630 IF SCRN(A-1,B)=ASC("o")THENGOTO 2900
640 IF SCRN(A-1,B)=ASC("c")THEN Q=Q+10:PR=PR+1
650 IF PR=125 THEN GOTO 50
660 PLOT A,B,"  ":PLOT C,D,"  "
670 A=A-2:C=C-2
680 PLOT A,B,Q$:PLOT C,D,W$
690 GH1=INT(RND(1)*4):IFGH1=0THEN GOSUB 1720
700 IFGH1=1THEN GOSUB 1810
710 IF GH1=2THEN  GOSUB 1900
720 IF GH1=3THEN GOSUB 1990
730 GH2=INT(RND(1)*4):IFGH2=0THEN GOSUB 2080
740 IF GH2=1THEN GOSUB 2170
750 IF GH2=2THEN GOSUB 2260
```

```
760 IF GH2=3THEN GOSUB 2350
770 PLOT 25,0,"T I M E "
780 TIME=TIME-1
790 TIME$=LEFT$(STR$(TIME),3)
800 PLOT 35,0,TIME$
810 PLOT 10,0,"SCORE :"
820 PLOT 20,0,STR$(Q)
830 A$=KEY$
840 IFA$=CHR$(9)THENGOTO 880
850 IFA$=CHR$(10)THEN GOTO 1160
860 IFA$=CHR$(11)THEN GOTO 1440
870 GOTO 600
880 REM **** RIGHT ****
890 IF SCRN(C+2,D)=ASC("h")THEN RETURN
900 IF SCRN (C+2,D)=ASC("a")THEN RETURN
910 IF SCRN(C+2,D)=ASC("p")THENGOTO 2900
920 IF SCRN(A+2,B)=ASC("b")THEN Q=Q+10:PR=PR+1
930 IF PR=125 THEN GOTO 50
940 PLOT A,B,"  ":PLOT C,D,"  "
950 A=A+2:C=C+2
960 PLOT A,B,Q$:PLOT C,D,W$
970 GH1=INT(RND(1)*4):IFGH1=0THEN GOSUB 1720
980 IFGH1=1THEN GOSUB 1810
990 IF GH1=2THEN  GOSUB 1900
1000 IF GH1=3THEN GOSUB 1990
1010 GH2=INT(RND(1)*4):IFGH2=0THEN GOSUB 2080
1020 IF GH2=1THEN GOSUB 2170
1030 IF GH2=2THEN GOSUB 2260
1040 IF GH2=3THEN GOSUB 2350
1050 PLOT 25,0,"T I M E "
1060 TIME=TIME-1
1070 TIME$=LEFT$(STR$(TIME),3)
1080 PLOT 35,0,TIME$
1090 PLOT 10,0,"SCORE :"
1100 PLOT 20,0,STR$(Q)
1110 A$=KEY$
1120 IFA$=CHR$(8)THENGOTO 600
1130 IFA$=CHR$(10)THEN GOTO 1160
1140 IFA$=CHR$(11)THEN GOTO 1440
1150 GOTO 880
1160 REM **** DOWN *****
1170 IF SCRN(C,D+1)=ASC("f") THEN RETURN
1180 IF SCRN(C,D+1)=ASC("a") THEN RETURN
1190 IF SCRN(C,D+1)=ASC("n")GOTO 2900
1200 IF SCRN(C,D+1)=ASC("b") THEN Q=Q+10:PR=PR+1
1210 IF PR=125 THEN GOTO 50
1220 PLOT A,B,"  ":PLOT C,D,"  "
1230 B=B+2:D=D+2
1240 PLOT A,B,Q$:PLOT C,D,W$
1250 GH1=INT(RND(1)*4):IFGH1=0THEN GOSUB 1720
1260 IFGH1=1THEN GOSUB 1810
1270 IF GH1=2THEN  GOSUB 1900
1280 IF GH1=3THEN GOSUB 1990
1290 GH2=INT(RND(1)*4):IFGH2=0THEN GOSUB 2080
1300 IF GH2=1THEN GOSUB 2170
1310 IF GH2=2THEN GOSUB 2260
1320 IF GH2=3THEN GOSUB 2350
1330 PLOT 25,0,"T I M E "
1340 TIME=TIME-1
1350 TIME$=LEFT$(STR$(TIME),3)
1360 PLOT 35,0,TIME$
1370 PLOT 10,0,"SCORE :"
1380 PLOT 20,0,STR$(Q)
1390 A$=KEY$
1400 IFA$=CHR$(8)THENGOTO 600
1410 IFA$=CHR$(09)THEN GOTO 880
1420 IFA$=CHR$(11)THEN GOTO 1440
1430 GOTO 1160
1440 REM **** UP *******
1450 IF SCRN(A,B-1)=ASC("h") THEN RETURN
1460 IF SCRN(A,B-1)=ASC("a") THEN RETURN
1470 IF SCRN(A,B-1)=ASC("p")THENGOTO 2900
1480 IF SCRN( A,B-1)=ASC("d") THEN Q=Q+10:PR=PR+1
1490 IF PR=125 THEN GOTO 50
1500 PLOT A,B,"  ":PLOT C,D,"  "
1510 B=B-2:D=D-2
1520 PLOT A,B,Q$:PLOT C,D,W$
1530 GH1=INT(RND(1)*4):IFGH1=0THEN GOSUB 1720
1540 IFGH1=1THEN GOSUB 1810
1550 IF GH1=2THEN  GOSUB 1900
1560 IF GH1=3THEN GOSUB 1990
1570 GH2=INT(RND(1)*4):IFGH2=0THEN GOSUB 2080
1580 IF GH2=1THEN GOSUB 2170
1590 IF GH2=2THEN GOSUB 2260
1600 IF GH2=3THEN GOSUB 2350
1610 PLOT 25,0,"T I M E "
1620 TIME=TIME-1
1630 TIME$=LEFT$(STR$(TIME),3)
1640 PLOT 35,0,TIME$
1650 PLOT 10,0,"SCORE :"
1660 PLOT 20,0,STR$(Q)
1670 A$=KEY$
1680 IFA$=CHR$(8)THENGOTO 600
1690 IFA$=CHR$(9) THEN GOTO 880
1700 IFA$=CHR$(10)THEN GOTO 1160
1710 GOTO 1440
1720 REM **** GHOST - LEFT ****
1730 IF SCRN(E-1,F)=ASC("a")THEN RETURN
1740 IF SCRN(E-1,F)=ASC("g")THENRETURN
1750 IF SCRN(E-1,F)=ASC("k")THENGOTO2900
1760 PLOT E,F,"  ":PLOT G,H,"  "
1770 E=E-2:G=G-2
1780 PLOT E,F,E$:PLOT G,H,R$
1790 PLOT E+2,F,"bc":PLOT G+2,H,"de"
1800 RETURN
1810 REM **** GHOST - RIGHT ****
1820 IF SCRN(E+2,F)=ASC("a")THENRETURN
1830 IF SCRN(E+2,F)=ASC("f")THENRETURN
1840 IF SCRN(E+2,F)=ASC("j")THENGOTO2900
1850 PLOT E,F,"  ":PLOT G,H,"  "
1860 E=E+2:G=G+2
1870 PLOT E,F,E$:PLOTG,H,R$
1880 PLOT E-2,F,"bc":PLOT G-2,H,"de"
1890 RETURN
1900 REM **** GHOST - DOWN  ****
1910 IF SCRN(G,H+1)=ASC("a")THENRETURN
1920 IF SCRN(G,H+1)=ASC("f")THENRETURN
1930 IF SCRN(G,H+1)=ASC("j")THENGOTO2900
1940 PLOT E,F,"  ":PLOT G,H,"  "
1950 F=F+2:H=H+2
1960 PLOT E,F,E$:PLOT G,H,R$
1970 PLOT E,F-1,"de":PLOT E,F-2,"bc"
1980 RETURN
1990 REM **** GHOST - UP    ****
2000 IF SCRN(E,F-1)=ASC("a")THENRETURN
2010 IF SCRN(E,F-1)=ASC("h")THENRETURN
2020 IF SCRN(E,F-1)=ASC("l")THENGOTO2900
2030 PLOT E,F,"  ":PLOT G,H,"  "
2040 F=F-2:H=H-2
2050 PLOT E,F,E$:PLOT G,H,R$
2060 PLOT G,H+1,"bc":PLOT G,H+2,"de"
2070 RETURN
2080 REM **** GHOST-2- LEFT ******
2090 IFSCRN(I-1,J)=ASC("a")THENRETURN
2100 IFSCRN(I-1,J)=ASC("g")THENRETURN
2110 IFSCRN(I-1,J)=ASC("k")THENGOTO2900
2120 PLOT I,J,"  ":PLOT K,L,"  "
2130 I=I-2:K=K-2
2140 PLOT I,J,E$:PLOTK,L,R$
2150 PLOT I+2,J,"bc":PLOT K+2,L,"de"
2160 RETURN
2170 REM **** GHOST-2- RIGHT *****
2180 IFSCRN(I+2,J)=ASC("a")THENRETURN
2190 IFSCRN(I+2,J)=ASC("f")THENRETURN
2200 IFSCRN(I+2,J)=ASC("j")THENGOTO2900
2210 PLOT I,J,"  ":PLOT K,L,"  "
2220 I=I+2:K=K+2
2230 PLOT I,J,E$:PLOT K,L,R$
2240 PLOT I-2,J,"bc":PLOT K-2,L,"de"
2250 RETURN
2260 REM **** GHOST-2- DOWN ******
2270 IF SCRN(K,L+1)=ASC("a")THENRETURN
2280 IFSCRN(K,L+1)=ASC("f")THENRETURN
2290 IFSCRN(K,L+1)=ASC("j")THENGOTO2900
2300 PLOT I,J,"  ":PLOT K,L,"  "
2310 J=J+2:L=L+2
2320 PLOT I,J,E$:PLOT K,L,R$
2330 PLOT I,J-1,"de":PLOT I,J-2,"bc"
2340 RETURN
2350 REM **** GHOST-3- UP ********
2360 IF SCRN(I,J-1)=ASC("a")THENRETURN
2370 IF SCRN(I,J-1)=ASC("h")THENRETURN
2380 IF SCRN(I,J-1)=ASC("l")THEN GOTO2900
2390 PLOT I,J,"  ":PLOT K,L,"  "
2400 J=J-2:L=L-2
2410 PLOT I,J,E$:PLOT K,L,R$
2420 PLOT K,L+1,"bc":PLOT K,L+2,"de"
2430 RETURN
2440 END
2450 REM ***** REDEFINE ********
2460 FOR A=0TO135:READD:POKE 46080+A+(8*ASC("a")),D:NEXTA
2470 REM ****** DATA DUMP *******
2480 DATA 63,33,33,33,33,33,33,63
2490 DATA 0,0,0,0,0,0,3,3
2500 DATA 0,0,0,0,0,0,48,48
2510 DATA 3,3,0,0,0,0,0,0
2520 DATA 48,48,0,0,0,0,0,0
2530 DATA 63,63,63,63,63,63,63,63
2540 DATA 63,63,63,63,63,63,63,63
2550 DATA 63,63,63,63,63,63,63,63
2560 DATA 63,63,63,63,63,63,63,63
2570 DATA 3,3,15,15,63,63,51,51
2580 DATA 48,48,60,60,63,63,15,15
2590 DATA 51,51,63,63,12,12,60,60
2600 DATA 15,15,63,63,12,12,15,15
2610 DATA 3,3,15,12,60,60,63,63
2620 DATA 48,48,60,36,39,39,63,63
2630 DATA 63,48,32,48,63,15,15,3
2640 DATA 63,3,1,3,63,60,60,48
2650 RETURN
2660 REM ******* INSTRUCTIONS ******
2670 CLS
2680 PLOT 1,1,"THE OBJECT OF THE GAME IS TO EAT THE"
2690 PLOT 1,3,"DOTS BEFORE THE TIME RUNS OUT OR    "
2700 PLOT 1,5,"BEFORE THE GHOSTS EAT YOU . "
2710 PLOT 1,8," YOU ARE ORCMAN............. jk"
2720 PLOT 1,9,"                             lm"
2730 PLOT 1,11," THE GHOSTS ARE ............no"
2740 PLOT 1,12,"                            pq"
2750 PLOT 1,14," GUIDE ORCMAN THROUGH THE MAZE BY   "
2760 PLOT 1,16," THE CURSOR KEYS "
2770 PLOT 1,18," THE GHOSTS LAY DOTS WHILE RUNING "
2780 PLOT 3,20," REMEMBER YOU HAVE ONLY 23 ZOIDS "
2790 PLOT 3,22," TO EAT THE DOTS
2800 PLOT 5,24," PRESS ANY KEY "
2810 PLOT 4,24,0:PLOT 3,24,18:PLOT 20,24,0
2820 FOR A=1TO20
2830 READ B
2840 PLAY 7,0,0,0
2850 MUSIC 1,3,B,10:WAIT20
2860 PLAY0,0,0,0 :NEXTA
2870 DATA 3,5,1,3,5,6,8,8,6,5,3,6,5,3,1,5,6,8,10,6
2880 GET V$
2890 RETURN
2900 REM ****** FINALE ********
2910 CLS
2920 PLOT 10,13," YOU ARE DEAD"
2930 PLOT 10,14," YOU ARE DEAD"
2940 PLOT 10,13,10
2950 PLOT 10,14,10
2960 FORX=1TO21
2970 READ A,B
2980 PLAY 7,0,0,0
2990 MUSIC 1,3,A,10
3000 WAITB*10
3010 NEXTX
3020 DATA 1,3,3,3,5,3,6,3,1,3,1,4,6,3,8,3,9,3,5,3,6,3,1,3,6,1,8,1,1
0,1
3030 DATA 1,4,5,3,6,3,10,3,10,3,6,3
3040 PLAY 0,0,0,0
3050 PLOT 10,23," DO YOU WANT ANOTHER GO ?"
3060 INPUT V$
3070 IF V$="Y" THEN RUN
3080 END
```

S W Lucas

SCHOOL OF DEATH

You have to find the hidden 'treasure' in your brother's school, but believe me, school was never like this in my day!

.... COULD THIS BE MY FIRST CLUE FRED ASKED HIMSELF !

Hello there. My name is Fred and I have been sent to my younger brother's school just before the caretaker is locking up to go home for the night. My mother has given me strict instructions to find the hidden five valuable items, take them to the car park and call her to come and collect them.

There are many dangers to be faced in the school and the task will not be easy.

WORDS UNDERSTOOD

E, W, N, S, CLIMB, JUMP, DOWN, SWIM, GET, TAKE, GRAB, USE, INSERT, SCORE, THROW, EXAMINE, INVENTORY, HELP, PRAY, DROP, LEAVE, QUIT, LOOK, PUSH, SHOOT, ATTACK, KILL, HIT, WAR, GIVE, BUY, READ, EAT, LOAD, CLOAD, SEARCH, KISS, SWEARING.

Any of these words can be used to instruct the program and operate the game throughout.

CONVERSION CLUES

Most of the program uses fairly standard BASIC. It should be possible to convert the program fairly easily to run on most machines:

PAPER and INK only change the colour of the display and may be left out altogether. ZAP, PING, SHOOT and EXPLODE produce predefined sounds and may be replaced by almost any short sound effects (or left out altogether).

WAIT could be replaced with a For/Next loop to introduce a short time delay.

REPEAT/UNTIL can easily be replaced by a FOR/NEXT loop.

VARIABLES USED

Integer variables are used to save memory space.

P% = current location.

X$(X) = messages.

V$(X) = what you are carrying.

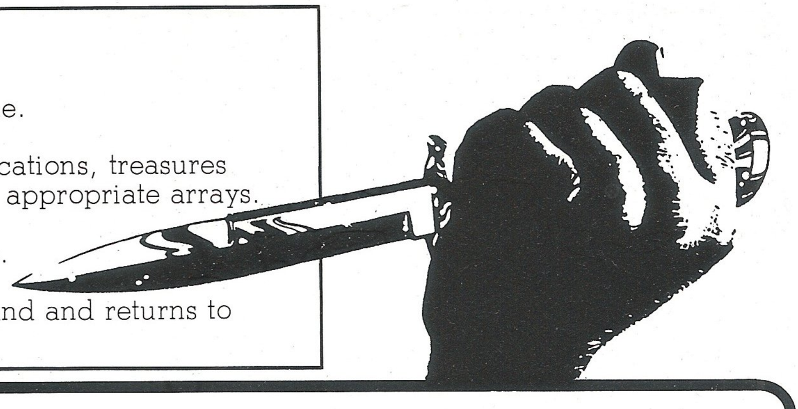
S%(X,Y) = where you can go from each location.

AN = FRE("") in line 102 is a garbage collection routine not necessary in some versions of BASIC.

Z% = start location.

RUNdown

Line	Effect
40	Calls the TITLES subroutine.
60	Sets pointer for variables.
70-80	Calls subroutine to read locations, treasures and words understood into appropriate arrays.
100-160	Displays current location.
180	Displays where you can go.
250	Calls for input.
260-470	Acts upon inputted command and returns to display at line 100.



```
1 REM ++ AN ADVENTURE GAME WRITTEN FOR THE 48 K ORIC 1

2 REM ++ BY S.W. LUCAS 1983

--

10 TEXT:PAPER2:INK0

20 C%=0:Y%=0:X%=0:J%=0:K%=0:P%=2:M%=0:Z%=2

30 DIMQ$(53),G$(53),S%(53,4),V$(3),B%(60,1),N$(69),N%(69),X$(61)

40 GOSUB2720

60 RESTORE

70 GOSUB490

75 GOSUB2290

80 CLS

100 PRINT:PRINT:PRINT"I AM":PRINTQ$(P%):A$=""

101 PRINTCHR$(8)

102 AN=FRE("")

110 IFS%(P%,1)>0THENA$="NORTH"

120 GOSUB2770

130 IFS%(P%,2)>0ANDLEN(A$)>0THENA$=A$+",SOUTH"

131 IFS%(P%,2)>0ANDLEN(A$)=0THENA$="SOUTH"

140 IFS%(P%,3)>0ANDLEN(A$)>0THENA$=A$+",EAST"

141 IFS%(P%,3)>0ANDLEN(A$)=0THENA$="EAST"

150 IFS%(P%,4)>0ANDLEN(A$)>0THENA$=A$+",WEST"

151 IFS%(P%,4)>0ANDLEN(A$)=0THENA$="WEST"

160 IFS%(P%,1)=0ANDS%(P%,2)=0ANDS%(P%,3)=0ANDS%(P%,4)=0THENA$="NOWH

ERE AT ALL"

161 IFB%=1THENA$="DOWN":S%(P%,1)=0:S%(P%,2)=0:S%(P%,3)=0:S%(P%,4)=0

180 PRINT" I CAN GO :-":PRINTA$

200 E=0:FORT=1TO51:PP%=0:IFB%(T,1)=P%THENPP%=1

210 IFPP%=1THEN230

220 NEXT:GOTO250

230 IFE=0THENPRINT:PRINT"THAT LOOKS LIKE"

240 PRINTG$(T):E=E+1:GOTO220

250 PRINT:INPUT"WHAT SHOULD I DO NOW";Z$

260 B$=LEFT$(Z$,1)

261 C$=LEFT$(Z$,3)

270 CLS:IF LEFT$(Z$,2)="GO"THENPRINT"JUST USE THE FIRST LETTER":GOT

O100

271 PRINT:PRINT:PRINT"HOLD ON A SEC!":PRINT:PRINT

280 IFB$="N"ANDS%(P%,1)<>0THENP%=S%(P%,1):GOTO100

320 IFB$="W"ANDS%(P%,4)<>0THENP%=S%(P%,4):GOTO100

330 B$=LEFT$(Z$,2)

340 IFB$="S"ANDMID$(Z$,2,1)<>"H"ANDS%(P%,2)<>0THENP%=S%(P%,2):GOTO1

00

350 IFB$="SC"THENGOSUB2770:PRINT"YOU HAVE SCORED ";A%;" OUT OF 6":G

OTO100

360 IFB$="TH"THEN3040

370 IFC$="E"ANDS%(P%,3)<>0ANDMID$(B$,2,1)<>"X"THENP%=S%(P%,3):GOTO1

00
```

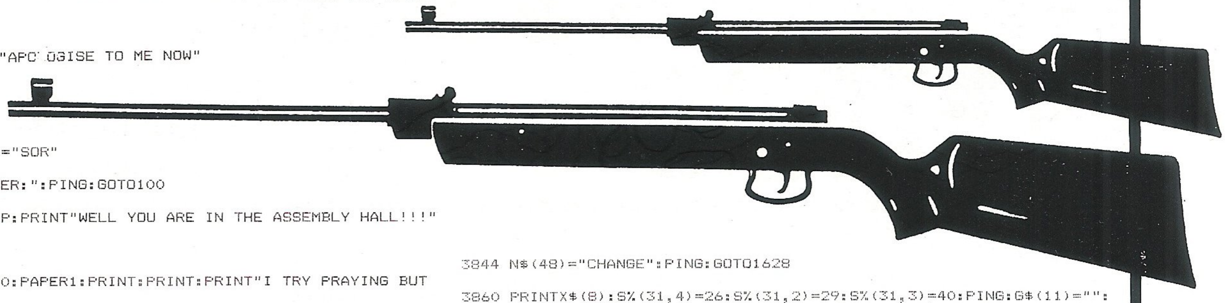
```
380 IFB$="GE"ORB$="TA"ORB$="GR"THEN1550

381 IFC$="CLI"THEN3550

385 IFB$="RE"THEN2940

390 IFB$="FU"ORB$="PI"THEN2840

391 IFC$="DOW"THEN3580

395 IFB$="HE"THEN2270

397 IFC$="SEA"THEN4100

400 IFB$="DR"ORB$="LE"ORC$="GIV"THEN1790

401 IFC$="PHO"THEN3900

405 IFB$="QU"THEN3610

410 IFC$="LOO"THEN10000

411 IFC$="KIS"THEN3930

415 IFB$="PR"THEN2890

416 IFC$="LOA"ORC$="CLO"THEN4000

420 IFB$="PU"THEN3010

421 IFC$="EAT"THEN3700

425 IFB$="AT"ORB$="HI"ORB$="KI"THENZAP:GOTO1990

430 IFB$="EX"THEN2060

431 IFC$="SWI"THENCLS:ZAP:PRINT:PRINT:PRINT"SWIM? HERE ? What do yo

u think I am?":GOTO100

435 IFC$="USE"ORC$="INS"THEN2230

440 IFC$="WEA"THEN3060

445 IFC$="JUM"THEN3500

450 IFC$="INV"THEN2150

454 IFC$="BUY"THEN3800

460 K$=LEFT$(B$,1)

461 IFLEFT$(A$,3)<>"SOR"THENLL=1 ELSELL=0

465 IFB$<>""ANDLL=1ANDK$<>"S"ANDK$<>"N"ANDK$<>"E"ANDK$<>"W"THENPRIN

TX$(60)

470 GOTO100

480 END

490 FORH=1TO51:FORD=1TO4

500 READS%(H,D):NEXT

520 READQ$(H):NEXT

530 FORH=1TO51

540 READG$(H),B%(H,1):NEXT

550 FORH=1TO69

560 READN$(H),N%(H)

565 NEXT

570 RETURN

580 DATA2,0,0,0,"IN A FISH AND CHIP SHOP"

590 DATA4,0,3,0,"ON THE PAVEMENT. THE SCHOOL IS NEARBY"

600 DATA0,0,5,2,"IN THE CAR PARK. IT IS FULL OF CARS"

610 DATA0,2,0,0,"IN THE SWEETSHOP"

620 DATA6,0,0,3,"BY THE SCHOOL'S MAIN ENTRANCE"

630 DATA11,5,13,7,"IN THE ENTRANCE HALL"

640 DA~A8,0,6,0,"IN THE DINING HALL"
```

```
650 DATA0,7,0,42,"IN THE KITCHENS"
660 DATA0,0,10,0,"IN THE BOYS CHANGING ROOMS"
670 DATA43,11,17,9,"IN A GYMNASIUM"
680 DATA10,6,0,49,"IN A COVERED PASSAGEWAY"
690 DATA0,13,0,0,"IN THE DEPUTY HEAD'S OFFICE"
700 DATA12,15,14,6,"IN THE MAIN CORRIDOR"
710 DATA17,16,22,13,"IN THE~MAIN CORRIDOR"
720 DATA13,0,16,0,"IN THE GENERAL OFFICE"
730 DATA14,28,0,15,"IN THE HEADMASTER'S OFFICE"
740 DATA18,14,0,10,"IN THE ASSEMBLY HALL"
750 DATA47,17,0,0,"ON THE STAGE"
760 DATA0,20,34,0,"IN THE PLAYGROUND"
770 DATA19,22,23,21,"IN A CORRIDOR"
780 DATA0,0,20,0,"IN A CLASSROOM"
790 DATA20,24,32,14,"IN A CORRIDOR"
800 DATA0,0,0,20,"IN A CLASSROOM"
810 DATA22,0,25,0,"IN A CLASSROOM"
820 DATA0,0,0,24,"IN THE MATHS STOCKROOM"
830 DATA0,0,31,0,"IN THE MUSIC PRACTICE ROOM"
840 DATA0,44,29,28,"IN THE SCHOOL LIBRARY"
850 DATA0,0,27,16,"IN THE REFERENCE SECTION OF THE       LIBRARY."
860 DATA31,30,41,27,"IN A LONG CORRIDOR"
870 DATA29,0,41,0,"AT THE DOOR OF THE STAFFROOM. "
880 DATA32,0,0,0,"IN A CORRIDOR"
890 DATA33,31,48,22,"IN A CORRIDOR"
900 DATA0,32,0,0,"IN THE GIRL'S TOILETS"
910 DATA0,35,0,19,"IN THE COMPUTER ROOM"
920 DATA34,37,0,51,"IN A PHYSICS LABORATORY"
930 DATA0,0,0,37,"IN A CHEMISTRY STOCK ROOM"
940 DATA35,38,0,0,"IN A PREP ROOM"
950 DATA37,0,0,0,"In a quadrangle. It is so icy that I  can't go So
uth!"
960 DATA38,0,0~0,"In the Common Room"
970 DATA0,0,0,31,"IN THE BOYS TOILETS"
980 DATA0,0,0,30,"IN THE STAFFROOM. THERE IS A TELEPHONEHERE."
990 DATA0,0,8,0,"BY THE DUSTBINS"
1000 DATA0,10,0,0,"IN THE GIRLS CHANGING ROOMS."
1010 DATA27,0,0,0,"IN A LANGUAGE LABORATORY"
1020 DATA0,19,0,46,"IN THE WORKSHOPS"
1030 DATA0,0,45,0,"IN THE ART ROOM. IT IS FULL OF        PAINTINGS.
"
1031 DATA0,18,0,0,"IN A STAGE LIGHTING BOX"
1032 DATA0,0,0,32,"IN A COOKERY ROOM"
1033 DATA50,0,11,0,"IN THE CARETAKER'S ROOM"
1034 DATA0,49,0,0,"IN THE BOILER ROOM"
1035 DATA0,0,35,0,"IN A BIOLOGY LABORATORY"
1040 RETURN
1050 DATA"A TELEPHONE",41
1060 DATA"SOME HISTORY HOMEWORK.",21
1070 DATA"A DRESS",43
1080 DATA"A PAIR OF HEADPHONES",44
1090 DATA"A PIECE OF MUSIC",26
1100 DATA"A GANG OF THUGS BEATING UP AN OLD LADY",2
1110 DATA"A BAR OF CHOCOLATE",4
1120 DATA"A STRAY DOG",19
1130 DATA"A SCHOOL DINNER",7
1140 DATA"A VAN DE GRAAFF GENERATOR",35
1150 DATA"THE SCHOOL BULLY. HE DEMANDS A BRIBE",31
1160 DATA"A PILE OF EXCERCISE BOOKS",25
1170 DATA My valuable#WATCH# in the lost           property box",12
1180 DATA"My brother's solid #SILVER# pencil     case",23
1190 DATA"My Girlfriend",38
1200 DATA"A PACKET OF CIGARETTES",4
1210 DATA"A LABORATORY TECHNICIAN",37
1220 DATA"A CAGE OF GERBILS",51
1230 DATA"A LARGE MASTER SWITCH",47
1240 DATA"A PAINTING",46
1250 DATA"A HAMMER",49
1260 DATA"A BOTTLE OF HYDROCHLORIC ACID",36
1270 DATA"A RAW EGG",48
1280 DATA"A BAKING TRAY",48
1290 DATA"A STEEL BAR",45
1300 DATA"My #ORIC# computer",39
1310 DATA"A TOILET ROLL",40
1320 DATA"A PAIR OF SHORTS",9
1330 DATA"A ROPE",10
1340 DATA"A WALL BAR",10
1350 DATA"A BAR OF SOAP",33
1360 DATA"SOME CHIPS",1
1370 DATA"A £5 NOTE ON THE FLOOR",3
1380 DATA"A CRICKET TROPHY",6
1390 DATA"A JAR OF SODIUM CHLORIDE",36
1400 DATA"A CANE",16
1410 DATA"A REGISTER",15
1411 DATA"A SCHOOL REPORT",12
1412 DATA"AN IRATE TEACHER",23
1413 DATA" MAGAZINE",27
1414 DATA"AN ENCYCLOPEDIA",28
1415 DATA"MY MATHS WORK",30
1416 DATA"A CLOCK",14
1417 DATA"A TABLE",18
1418 DATA"A DESK",21
1419 DATA"A PACKET OF CRISPS",4
1420 DATA"A GOLD FOUNTAIN PEN",24
1421 DATA"A PILE OF RUBBISH",42
1422 DATA"A TAPE RECORDER",44
1423 DATA"A MICROCOMPUTER",34
1424 DATA"A CASSETTE TAPE",24
1430 DATATELEPHONE,1,HOMEWORK,2,HISTORY,2
1435 DATA"DRESS",3,"HEADPRETURNONES",4,"MUSIC",5,"PIECE",5
1440 DATATHUGS,6,LADY,6,CHOCOLATE,7,BAR,7
1445 DATADOG,8,DINNER,9,GENERATOR,10,VAN DE GRAAFF,10
1450 DATABULLY,11,EXCERCISE,12,WATCH,13,BOX,13
1455 DATAPENCILCASE,14,SILVER,14,GIRLFRIEND,15
1460 DATACIGARETTES,16,PACKET,16,TECHNICIAN,17
1465 DATACAGE,18,GERBILS,18,SWITCH,19,PAINTING,20
1470 DATAHAMMER,21,ACID,22,HYDROCHLORIC,22,BOTTLE,22
1475 DATAEGG,23,BAKING,24,TRAY,24
1480 DATASTEEL,25,BAR,25,ORIC1,26,COMPUTER,26
1485 DATATOILET,27,ROLL,27,SHORTS,28,ROPE,29
1490 DATAWALLBAR,30,SOAP,31,CHIPS,32,NOTE,33
1495 DATATROPHY,34,SALT,35,CANE,36,REGISTER,37
1500 DATAREPORT,38,TEACHER,39,MAGAZINE,40,ENCYCLOPEDIA,41
1505 DATAMATHS,42,CLOCK,43,TABLE,44,DESK,45,DRAW,45,CRISPS,46
1510 DATAPEN,47,RUBBISH,48,RECORDER,49,TAPE RECORDER,49,MICROCOMPUT
ER,50
1515 DATABBC,50,CASSETTE,51
1550 GOSUB2640
```

```
1560 IFL%=1THEN1580
1570 GOTO250
1580 E%=0
1590 FORH=1TO51:IFB%(H,1)=P%ANDB%(N%(R),1)=P%THENE%=1
1600 NEXT
1610 IFE%=0THEN250
1620 CLS
1621 IFR=13THENAA=1
1622 IFP%=2ANDR=8ANDAD<>1THENPRINTX$(3):PING:GOTO100
1623 IFP%=2ANDR=9ANDAD<>1THENPRINTX$(4):EXPLODE:GOTO3600
1624 IFP%=4AND(R=10ORR=11ORR=23ORR=24ORR=62)THENPRINTX$(3):ZAP:GOTO
100
1625 IFR=47THENAC=1
1626 IFP%=19ANDR=12ANDAE<>1THENPRINTX$(6):SHOOT:WAIT10:GOTO100
1627 IFP%=19ANDR=12ANDAE=1THENPRINTX$(7)
1628 IFR=10ORR=11THENAB=1
1629 IFR=48THENAF=1
1630 IFP%=3ANDR=48THENG$(33)="A £5 NOTE"
1631 IFP%=41ANDR=1THENPRINT"VANDAL!!: I'LL HAVE NO PART IN SUCH   A
CTION":GOTO100
1632 IFP%=31ANDR=16THENPRINT"WHAT DO YOU TAKE ME FOR? AN IDIOT?":ZA
P:GOTO100
1633 IFP%=6ANDR=49THENPRINT"YOU MIGHT BE A THIEF, BUT I'M NOT!":PIN
G:GOTO100
1634 IFP%=27ANDR=55THENPRINTX$(16):SHOOT:WAIT20:PRINTX$(17):SHOOT:P
%=16:GOTO100
1635 IFP%=36AND(R=31ORR=32ORR=33)THENGOTO3870
1636 IFP%=37ANDR=25THENPRINT:PRINT:PRINT:PRINT"What do you think th
is is? a GAME?":PING:GOTO100
1637 IFP%=38ANDR=22ANDAG<>1THENPRINTX$(24):ZAP:G$(15)="":GOTO100
1638 IFP%=34AND(R=67ORR=68)THENZAP:PRINT"I AM ARRESTED FOR STEALING
":GOTO3610
1639 IFP%=44AND(R=65ORR=66)THENAH=1
1640 IFP%=42ANDR=64ANDAI<>1THENPRINTX$(32):GOTO100
1641 IFP%=42ANDR=64THENPRINTX$(43):G$(48)="A KEY"
1642 IFP%=23AND(R=20ORR=21)ANDAJ<>1THENPRINTX$(33):ZAP:PAPER6:INK0:
GOTO100
1643 IFP%=23ANDR=54THENPRINT"Just how am I supposed to do that ?":G
OTO100
1644 IFP%=35AND(R=14ORR=15)THENPRINTX$(36):PRINTX$(37):ZAP:GOTO3610
1645 IFP%=21ANDR=60THENPRINT"I'm not stupid you know!":GOTO100
1646 IFP%=18ANDR=59THENPRINT"If you think I'm carrying that FORGET
IT":GOTO100
1647 IFP%=12ANDR=19THENPRINTX$(40):PING:GOTO100
1648 IFP%=12ANDR=18ANDAK<>1THENPRINTX$(41):PING:GOTO100
1649 IFP%=12ANDR=18THENG$(13)="My valuable £WATCH£"
1650 IFP%=33ORP%=43THENPRINTX$(45):GOTO3610
1651 IFP%=47ANDR=28THENPRINT"IMPOSSIBLE":PING:GOTO100
1740 E%=0
1750 FORD=1TO3:IFV$(D)=""THENV$(D)=G$(N%(R)):E%=1:D=5
1760 NEXT
1770 IFE%=0THENPRINT"SORRY. MY HANDS ARE FULL !":PING:GOTO250
1780 B%(N%(R),1)=0:PING:GOTO100
1790 GOSUB2640
1800 IFL%=1THEN1820
1810 PRINT"I HAVEN'T GOT A ";L$:ZAP:INK1:PAPER7:GOTO250
1820 E%=0
1830 FORD=1TO3:IFV$(D)=G$(N%(R))THENV$(D)="":E%=1
1840 NEXT
```

```
1850 IFE%=1THEN1870
1860 PRINT"I DON'T HAVE IT . WHO'S A SILLY PERSONTHEN?":PING:GOTO10
0
1870 B%(N%(R),1)=P%
1880 CLS
1881 IFAA=1ANDR=13THENAA=0
1882 IFAB=1AND(R=10ORR=11)THENAB=0
1883 IFAC=1ANDR=47THENAC=0
1884 IFP%=2ANDR=12THENPRINTX$(5):G$(6)="A #GOLD# PENDANT":S%(2,2)=1
:AD=1
1885 IFP%=2ANDR=12THENN$(8)="GOLD":N$(9)="PENDANT":EXPLODE:GOTO100
1886 IFP%=19ANDR=62THENAE=1:PRINTX$(7)
1887 IFAF=1ANDR=48THENAF=0
1888 IFP%=31AND(R=23ORR=24)THEN3860
1889 IFP%=37ANDR=30THENGOTO3910
1890 IFP%=38AND(R=10ORR=11)THENPRINTX$(27):G$(15)="A #PRESENT#":N$(
22)="PRESENT"
1891 IFP%=38AND(R=10ORR=11)THENAG=1:ZAP:PAPER5:INK4:PRINTX$(28):G$(
7)="":GOTO100
1892 IFP%=38ANDR=50THENPRINTX$(29):S%(38,2)=39:Q$(38)="In the quadr
angle"
1893 IFP%=38ANDR=50THENG$(35)="":PING:GOTO100
1894 IF(R=66ORR=67)THENAH=0
1895 IFP%=34ANDAH=1AND(R=65ORR=66)THENAH=2
1896 IFP%=23ANDR=57THENPRINTX$(34):AJ=1:PING:G$(39)="":GOTO100
1897 IFP%=23AND(R=53ORR=20RR=3)ANDAJ<>1THENPRINTX$(35):ZAP
1898 IFR=34THENPRINTX$(42):G$(23)="":EXPLODE:GOTO100
1980 GOTO100
1990 IFP%=2THENPRINT"I'm not doing that!":ZAP:GOTO100
1991 PRINT"ATTACK? you're joking":PING :GOTO100
2060 GOSUB2640
2070 IFP%=2ANDR=9THENPRINTX$(1):ZAP:GOTO250
2071 IFP%=2ANDR=8THENPRINTX$(2):GOTO250
2072 IFP%=37ANDR=25THENPRINTX$(20):PRINTX$(21):GOTO100
2073 IFP%=38ANDR=22ANDAG<>1THENPRINTX$(25):PING:GOTO100
2074 IFP%=34AND(R=67ORR=68)THENPRINTX$(30):GOTO100
2140 PRINT"I can't see anything special":GOTO100
2150 PRINT:PRINT:PRINT:PRINT"I HAVE:-"
2160 F%=0
2170 FORH=1TO3:IFV$(H)<>""THENPRINTV$(H):F%=1
2180 NEXT
2190 IFF%=0THENPRINT"NOT A SAUSAGE!!":ZAP:GOTO250
2200 PAPER3:INK0:PING:GOTO250
2230 CLS:PING:PAPER6:INK0:GOSUB2640
2231 IFP%=23ANDR=64THENPRINTX$(38):GOTO100
2235 IFP%=12 ANDR=64THENPRINTX$(39):AK=1:GOTO100
2236 PRINT"I'm afraid I just don't understand you":ZAP:GOTO100
2270 CLS:PING:PAPER6:INK4:
2271 IFP%=2THENPRINT:PRINT:PRINT:PRINT"THEY MUST BE FRIGHTENED OF S
OMETHING!"::PING:GOTO100
2272 IFP%=1THENPRINT:PRINT:PRINT:PRINT"THEY LOOK TASTY":PING:GOTO25
0
2273 IFP%=4THENPRINT:PRINT:PRINT:PRINT"CIGARETTES ARE BAD FOR THE H
EALTH!!!!":PING:GOTO100
2274 IFP%=7THENPRINT:PRINT:PRINT:PRINT"COULD THAT BE A RED HERRING
FOR DINNER?":ZAP:GOTO100
2275 IFP%=16THENPRINT:PRINT:PRINT:PRINT"I'D WATCH IT IF I WERE YOU!
!":GO~0100
2276 IFP%=30THENPRINT:PRINT:PRINT:PRINT"TELEPHONES CAN BE USEFUL!":
```

```
GOTO250
2277 IFP%=47THENPRINT:PRINT:PRINT:PRINT"PULL THE SWITCH. I DARE YOU
 !!":ZAP:GOTO250
2278 IFP%=26THENPRINT:PRINT:PRINT:PRINT"IT'S TOO NOISY HERE !!!":EX
PLODE:GOTO100
2279 IFP%=34THENPRINT"WHAT DO YOU DO WITH A MICRO?":PING:GOTO100
2280 IFP%=19THENPRINT:PRINT:PRINT:PRINT"I'D FIND SOMETHING QUICKLY"
:GOTO100
2281 IFP%=33THENPRINT:PRINT:PRINT:PRINT"WHAT AM I DOING IN HERE?":G
OTO100
2282 IFP%=35THENPRINT:PRINT:PRINT:PRINT"WATCH IT!! THOSE THINGS ARE
 DANGEROUS":EXPLODE:GOTO100
2283 IFP%=18THENPRINT:PRINT:PRINT:PRINT"DO YOU HAVE A SPEACH PREPAR
ED?":PING:GOTO250
2284 IFP%=27ORP%=28THENPRINT:PRINT:PRINT:PRINT"THERE MUST BE SOMETH
ING TO READ HERE!":GOTO100
2285 IFP%=36THENPRINT:PRINT:PRINT:PRINT"I DON'T KNOW ONE CHEMICAL F
ROM ANOTHER":GOTO100
2286 IFP%=52THENPRINT:PRINT:PRINT:PRINT"PHEW ! IT'S HOT HERE":PING:
GOTO100
2287 IFP%=53THENPRINT:PRINT:PRINT:PRINT"SOMETHING USEFUL HERE":PING
:GOTO100
2288 IFP%=45THENPRINT:PRINT:PRINT:PRINT"WHAT WOULD YOU DO WITH A ST
EEL BAR?:ZAP:GOTO100
2289 PRINT:PRINT:PRINT:PRINT"I'M NOT GOING TO SOLVE THIS FOR YOU
YOU KNOW !!!":ZAP:GOTO250
2290 REM READ COMMENTS
2300 X$(1)="THE LADY IS CRYING FOR HELP"
2310 X$(2)="THE THUGS LOOK NASTY. I WOULDN'T FANCYMY CHANCES"
2311 X$(3)="What do you take me for ? I'm not a   shoplifter you kn
ow."
2312 X$(4)="AS I MOVE IN TO HELP, THEY ATTACK ME"
2313 X$(5)="THE THUGS TAKE FRIGHT. THE LADY DROPS SOMETHING AND RUN
S AWAY"
2314 X$(6)="THE DOG BARKS AND RUNS AWAY. I THINK  IT THINKS THIS IS
 A GAME!"
2315 X$(7)="IT WOLFS SOME OF THEM. IT IS HAPPY AND I THINK IT WILL
COME"
2316 X$(9)="THE ASSISTANT TAKES MY MONEY AND GIVESME CHANGE"
2317 X$(8)="THE BULLY GRABS THEM AND RUSHES TO AN AREA BEHIND THE B
IKE SHEDS"
2318 X$(10)="O.K. MY MOTHER ANSWERS. SHE TELLS ME  TO MEET HER IN T
HE CAR PARK"
2319 X$(11)="My mother answers. I get some earache about not findin
g treasures"
2320 X$(12)="IN THE CAR PARK. MY MOTHER HAS BROUGHTTHE CAR TO COLLE
CT ME"
2321 X$(13)="I'M SPOILT FOR CHOICE HERE! I CAN'T   DECIDE WHERE TO
START"
2322 X$(14)="The encylopedia falls open at a page  about making sal
t"
2323 X$(15)="I'm not sure if this is a clue?, but  it's about E.T.
PHONE HOME"
2324 X$(16)="The librarian accuses me of stealing  and takes me to
the head"
2325 X$(17)="He gives me a caning and then rushes  away to answer t
he phone"
2326 X$(18)="AS I REACH FOR THE BOTTLE, IT FALLS TOTHE FLOOR AND BR
EAKS."
2327 X$(19)="The acid sprays all over me. I don't  like this !!! Yo
u fool!!"
2328 X$(20)="She is in the middle of making        something, but c
an't find"
2329 X$(21)="the hammer. She won't let me into the Chemistry lab"
2330 X$(22)="She smiles and thanks me for helping  her and says tha
t she's"
2331 X$(23)="sorry that she has to go."
2332 X$(24)="She is not pleased with me and storms off in a rage"
2333 X$(25)="She is very beautiful, but she is inabad mood"
2334 X$(26)="She slaps me across the face and      storms off in a
rage"
2335 X$(27)="She grabs the chocolate, throws a my  birthday present
 on the"
2336 X$(28)="floor and storms off, still in a bad  mood"
2337 X$(29)="As I scatter it around, the ice melts and I can safely
 go South"
2338 X$(30)="It's one of those grotty BBC micros   not an ORIC like
 mine!"
2339 X$(31)="There doesn't seem to be much here"
2340 X$(32)="If you think I'm carrying half a ton  of smelly rubbis
h FORGET IT"
2341 X$(33)="Sorry, the teacher has locked it in   a cupboard"
2342 X$(34)="The teacher takes it off me and opens the cupboard"
2343 X$(35)="That makes her MAD! 'WHAT DO I WANT   THAT FOR?' she s
houts"
2344 X$(36)="I grab hold of the Van de Graaff  ~  generator. I fee
l a "
2345 X$(37)="Bolt of electricity surge through my  body!!!!"
2346 X$(38)="WRONG KEY!"
2347 X$(39)="The key fits. I open the box"
2348 X$(40)="It's far too heavy, besides it's not  mine to take"
2349 X$(41)="The box is locked. I can't open it"
2350 X$(43)="The rubbish is everywhere"
2351 X$(42)="the egg breaks. What a mess! it is    everywhere!"
2352 X$(45)="'WHAT ARE YOU DOING IN HERE?' shouts   the teacher. I
am expelled"
2353 X$(46)="It has gone completely dark. I trip   over and break m
y neck"
2630 RETURN
2640 L$="":FORH=1TOLEN(Z$)
2650 IFMID$(Z$,H,1)=" "THENL$=RIGHT$(Z$,(LEN(Z$)-H)):H=H+60
2660 NEXT
2670 R=0
2680 L%=0:IFLEN(L$)<2THENRETURN
2690 FORH=1TO69
2695 IFLEFT$(N$(H),LEN(L$))=L$ THENL%=1:R=H
2700 NEXT
2710 RETURN
2720 CLS:PAPER3:INK4:ZAP:WAIT10:ZAP
2730 CLS
2740 PLOT4,5,"THE SCHOOL OF DEATH"
2750 PLOT10,15,"AN ADVENTURE GAME BY"
2760 PLOT12,20,"S.W. LUCAS:
2770 A%=0
2780 IFB%(6,1)=3THENA%=A%+1
2785 IFB%(15,1)=3THENA%=A%+1
2790 IFB%(26,1)=3THENA%=A%+1
2795 IFB%(14,1)=3THENA%=A%+1
2800 IFB%(13,1)=3THENA%=A%+1
```



```
2830 RETURN
2840 CLS:PAPER2:INKO:ZAP:PRINT:PRINT:PRINT:PRINT"HOW DARE YOU SPEAK
 TO ME LIKE THAT"
2841 PRINT:PRINT:PRINT"APC OGISE TO ME NOW"
2842 REPEAT
2843 INPUTZ$
2844 UNTIL LEFT$(Z$,3)="SOR"
2845 PRINT"THAT'S BETTER:":PING:GOTO100
2890 IFP%=17THENCLS:ZAP:PRINT"WELL YOU ARE IN THE ASSEMBLY HALL!!!"
:GOTO100
2891 IFP%=2THENZAP:INKO:PAPER1:PRINT:PRINT:PRINT"I TRY PRAYING BUT
IT DOESN'T HELP":GOTO100
2892 PING:PAPER2:INKO:PRINT:PRINT:PRINT"PRAYING MAY BE GOOD FOR THE
 SOUL BUT !!!":GOTO100
2940 GOSUB2640
2941 IFP%=27ANDR=55THENPRINTX$(15):PING:GOTO100
2942 IFP%=28ANDR=56THENPRINTX$(14):PING:GOTO100
2943 IFP%=27THENPRINTX$(13):PING:GOTO100
2944 PRINT"There's not much of interest to read  here":PING:GOTO100
3000 IFP%=10ANDB%=1THENPRINT:PRINT:PRINT:PRINT"I JUMP FROM ~HE TOP
AND BREAK MY NECK":GOTO3500
3010 IFP%<>47THENPRINT"not at the moment":GOTO100
3011 GOSUB2640
3015 IFR=28THENPRINTX$(46):PING:GOTO3610
3016 PRINT"PULL WHAT?":PING:GOTO~00
3040 GOSUB2640
3060 PRINT"Now who's being silly ! ":ZAP:GOTO100
3111 GOTO100
3500 IFP%=10ANDB%=1THENPRINT:PRINT:PRINT:PRINT"I JUMP FROM THE TOP
AND BREAK MY NECK":GOTO3600
3510 PRINT:PRINT:PRINT:PRINT"I CAN'T JUMP HERE!":PING:GOTO100
3550 IFP%=10THENQ$(10)="AT THE TOP OF A ROPE":B%=1:GOTO100
3560 PRINT"DON'T BE STUPID ALL THE TIME":GOTO100
3580 IFP%=10ANDB%=1THENPRINT"O.K. I CLIMB DOWN THE ROPE":S%(10,1)=4
3:S%(10,2)=11
3581 IFP%=10ANDB%=1THENS%(10,3)=17:S%(10,4)=9:Q$(10)="IN A GYMNASIU
M"
3582 IFP%=10THENB%=0:GOTO100
3583 PRINT"HOW CAN I GO DOWN?":PING:GOTO100
3600 PAPER1:INK4:PRINT:PRINT:PRINT:PRINT"I AM DEAD !"
3610 PRINT:PRINT"DO YOU WANT TO PLAY AGAIN";
3620 INPUTX$:IFLEFT$(X$,1)="Y"THENRUN
3630 END
3700 IFAA=1THENPRINT"I EAT THE RED HERRING. AAAGGGHHH!":SHOOT:GOTO3
600
3701 GOSUB2640
3702 IF(R=10ORR=11)ANDAB=1THENPRINT"DELICIOUS":PING:GOTO3710
3703 IFR=47ANDAC=1THENPRINT"THAT WAS DELICIOUS.":GOTO3710
3704 IFR=62THENPRINT"SORRY CRISPS GIVE ME INDIGESTION":PING:GOTO100
3705 PRINT:PRINT:PRINT"NOT AT THE MOMENT":GOTO100
3710 FORH=1TO3:IFV$(H)=G$(32)THENV$(H)="":G$(32)="":NEXT
3711 FORH=1TO3:IFV$(H)=G$(7)THENV$(H)="":G$(7)="":GOTO~00
3712 GOTO100
3800 IFAF<>1THENPRINT"WHAT WITH?":PING:GOTO100
3810 GOSUB2640
3820 IFP%=4AND(R=10ORR=11ORR=23ORR=24ORR=62)THEN3840
3830 PRINT:PRINT:PRINT"NOT AT THE MOMENT":PING:GOTO100
3840 PRINTX$(9)
3841 FORD=1TO3:IFV$(D)=G$(33)THENG$(33)="SOME CHANGE":V$(D)="SOME C
HANGE":NEXT
3844 N$(48)="CHANGE":PING:GOTO1628
3860 PRINTX$(8):S%(31,4)=26:S%(31,2)=29:S%(31,3)=40:PING:G$(11)="":
G$(16)=""
3861 GOTO100
3870 CLS:EXPLODE:PRINTX$(18):WAIT10:PING:PRINTX$(19):GOTO3600
3900 IFP%<>41THENPRINT"I'M NOT E.T. you know! I need a phone":PING:
GOTO100
3901 GOSUB2640
3902 IFLEFT$(L$,3)<>"HOM"THENPRINT"I CAN'T REMEMBER THE NUMBER":ZAP
:GOTO100
3903 IFA%<>5THENPRINTX$(11):ZAP:GOTO100
3904 PING:PRINTX$(10):GOTO5000
3905 Q%(3)=X$(12):PING:GOTO100
3910 X$(20)="":X$(21)="":PRINTX$(22):PRINTX$(23):PING
3911 G$(17)="":S%(37,3)=36:GOTO100
3930 GOSUB2640
3940 IFP%<>38THENPRINT"What is there here to kiss ??":ZAP:PAPER2:IN
KO:GOTO100
3950 IFR=22THENPRINTX$(26):PING:G$(15)="":GOTO100
3960 PRINT"Eh?":PING:GOTO100
4000 IFP%<>34THENPRINT"I can't load a tape at the moment!":ZAP:GOTO
100
4005 GOSUB2640
4006 IFAH<>2THENPRINT"I don't have a tape recorder!":GOTO100
4010 IFR<>69THENPRINT"As usual, the tape I have won't load":PING:GO
TO100
4020 PRINT"The program loads. I thought that it  going to be anothe
r great game"
4030 PRINT"from S.W.LUCAS":PING:PRINT"BUT it isn't."
4040 PRINT"It just displays the message:-"
4050 PRINT"Your task is to rescue the old lady   from the thugs, fi
nd the five"
4060 PRINT"items of treasure, phone home for yourparents to collect
 you and"
4070 PRINT"load your treasures into your parents car!!!"
4080 ZAP:PRINT:PRINT:PRINT"HINT ! What is the chemical that can  me
lt ice?":PING:GOTO250
4100 IFP%<>42THENPRINTX$(31):GOTO100
4110 CLS:PAPER7:INKO:ZAP:PRINT:PRINT:PRINT:PRINT:PRINT:WAIT10:PRINT
"In amongst all the rubbish, "
4120 PRINT:PRINT"I can see a KEY"
4130 G$(48)="A key in amongst the rubbish":AI=1:N$(64)="KEY"
4140 GOTO100
5000 PRINT:PRINT:PRINT:PRINT:PRINT"WELL DONE!!!":PRINT:PRINT"YOU HA
VE SOLVED THIS ADVENTURE!"
5001 PING:PRINT"Your parents are in the car park and  have collect
d your treasure"
5002 END
10000 HIRES~
20000 TEXT:GOTO100
```

S W Lucas

MONSTER HUNT

A great maze/ adventure game where you hunt the monsters, instead of the other way around!

Go monster hunting in our maze/adventure game for any Oric computer.

When played, a board of five across by four down is displayed on the screen. Hidden on the board are 20 monsters. These monsters occur in two types, each of which can occur in 5 different colours. You have only twenty turns to hunt down the monsters. You may enter the coordinates of two locations at a time. The monsters hiding in these locations will then be displayed

RUNdown

Line	Action
5	Turns cursor off.
6	Sets score to zero.
10	Dimensions the array A(X) which holds the monsters positions.
20-30	Clears screen and chooses foreground and background colours.
35	Defines Z$ as a blank string used to erase 1 line on screen.
40	Calls subroutine for instructions.
50-60	Sets number of turns left and defines colours.
90	Calls subroutine which fills the array A(X) with the monsters in a random position.

100	Calls routine to draw the board.
110-140	Defines characters (these are chosen from the alternate set!!).
150-160	Sets serial attributes on screen to alternate character set/standard set.
200-790	Main control loop:-repeats loop until SC (score) = 10 or E% = 1 all turns used.
210	Decrease number of turns left.
225	Turns to string.
226-227	Used to strip first character off string to overcome the bug in the STR$ function in ORIC's basic which adds the serial attribute for GREEN to front of string.
230-260	Plots messages at screen coordinates.
270-300	Waits for number from 1-6 being pressed.
310	Prints number.
320-350	Waits for letter A-D and converts this to a number by subtracting 64 from its ASCII code.
355-356	Sets variables for array and graphics locations.
357-360	Plots location.
361	Checks if already displayed.
362	Calls routine to plot graphics.
365-480	Routine for second location.
490-620	Waits for space bar to be pressed.
630-660	Plot blank strings to erase areas of screen.
680	Calls routine to erase screen if you have not matched pair of monsters (KK 1).
800	If score = 10 call routine to win.
810-870	Routine for losing.
880	Turn cursor on again.
1000-1200	Draws board by setting serial attributes in TEXT mode.
2000-2070	Fills array A(X) with random numbers from 1-20 with no two locations identical.
4000-4290	Instructions routine.
4102-4106	Music ? routine.
4120-4140 and 4260-4280	Wait for space bar.
5000-5500	Erase graphics already plotted.
6000-6120	Plots graphics for monster held in location ZZ.
7000-7100	Win routine.
7021-7026	Music routine.

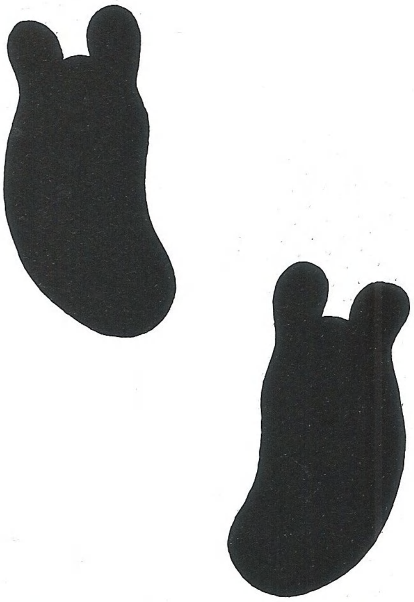
VARIABLES USED

SC =	Score.
E% =	Number of turns left.
A(X) =	Array which holds a random sequence of numbers for monsters locations.
T =	Used for music.
EE$	Holds string value of E%.
AA$,AB$, AC$,AD$=	Keys pressed for coordinates.
AA,AB,AC, AD=	Value of strings.
ZZ,ZF,ZG=	Location in array.
XX,XY,YX, YY=	Location for plotting graphics.
FA$=	Test for key pressed.
G$	Tests if you want another go.
X,A,Y,N	Counting loops.

on the screen until you press the space bar. Your task is to find the pairs of monsters which are identical in both shape and colour. If you do locate a matching pair, they will remain on the screen. You must try to locate all ten pairs of matching monsters before your twenty turns are up.

If you enter the coordinate of a location which is already being displayed and you will have wasted a turn. You must then press the space bar to get your next turn. You must tell the computer which square you want to view by typing in the coordinate in the form of number letter, eg. 2D not D2. The computer will not allow stupid entries.

The two monsters are built up out of the characters available in the alternate character set (TELETEXT graphic characters) and are thus not of a very detailed nature. It is quite straightforward to redefine these characters by POKING into the appropriate area of RAM to create monsters of your choice. These characters are held in the variables UA$,UB$,UC$,TA$, TB$,TC$ in lines 110-140.



CONVERSION CLUES

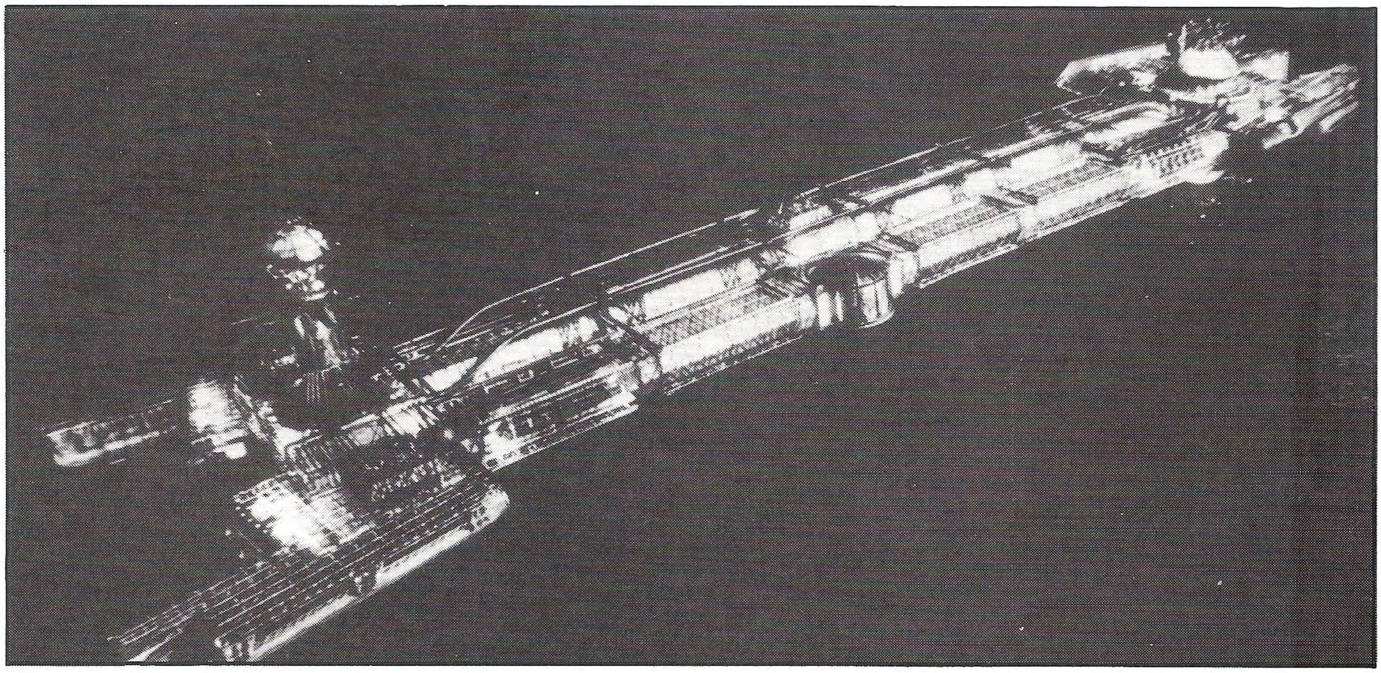
The program will work in a 16K Oric if all REM statements are removed and line 1 is replaced with TEXT-GRAB, which will release the area of memory set aside for high resolution graphics.

▶

```
EM ** MONSTER HUNT   **
2 REM ** a game to test your.memory **
3 REM ** for the ORIC1 **
4 REM ** <C> S.W. Lucas January 1984 **
5 PRINTCHR$(17) :REM ** TURN OFF CURSOR
6 SC=0 :REM SET SCORE TO ZERO AT START OF GAME
10 DIM A(20): REM ARRAY USED TO HOLD BOARD LOCATIONS
20 PAPER6:INK4
30 CLS
35 Z$="                                    "
40 GOSUB 4000
50 PAPER 6:INK4
60 E%=21:REM ** holds the number of turns left !
90 GOSUB2000:REM FILL ARRAY WHICH HOLDS BOARD LOCATIONS
100 GOSUB1000:REM DRAW THE BOARD
110 WA$="J[[":WB$="*L.":WC$="JKJ"
120 UA$="5J":UB$="J5":UC$="6I"
130 VA$="6 I":VB$=CHR$(34)+"\!":VC$="H£4"
140 TA$=" 5":TB$="J[" :TC$="JJ"
150 FOR P=2 TO 22:PLOT2,P,9:NEXT:REM ALTERNATE CHARACTER SET
160 FOR P=2 TO 22:PLOT29,P,8:NEXT:REM STANDARD CHARACTER SET
200 REPEAT
210 E%=E%-1
220 PLOT30,3,"goes left"
225 EE$=STR$(E%)
226 EE$=MID$(EE$,2)
227 EE$=EE$+" "
228 F=FRE("")
230 PLOT32,5,EE$
240 PLOT30,5,1
260 PLOT30,8,"loc. 1 ?"
270 REPEAT
280 GET AA$
290 AA=VAL(AA$)
300 UNTIL (AA>0 AND AA<6)
310 PLOT32,10,AA$
320 REPEAT
330 GET AB$
340 UNTIL AB$="A"ORAB$="B"ORAB$="C"ORAB$="D"
350 AB=ASC(AB$)-64
355 ZZ=(AB-1)*5+AA:ZF=ZZ:REM LOCATION IN.ARRAY
356 XX=AA*5-1:YY=AB*5-1:XF=XX:YF=YY :REM CO-ORDINATES FOR PLOTTING
GRAPHICS
357 PLOT 30,10,1
360 PLOT33,10,AB$:KK=0
361 IF A(ZZ)=0 THEN PLOT 29,1,"Silly!":KK=1:GOTO600
362 GOSUB 6000 : REM PLOT GRAPHICS
365 PLOT30,13,"loc.2 ?"
375 PLOT30,15,1
380 REPEAT
390 GET AC$
400 AC= VAL(AC$)
410 UNTIL (AC>0 AND AC<6)
420 PLOT32,15,AC$
430 REPEAT
440 GET AD$
450 UNTIL AD$="A"ORAD$="B"ORAD$="C"ORAD$="D"
460 AD=ASC(AD$)-64
470 PLOT33,15,AD$
475 XX=AC*5-1:YY=AD*5-1:XG=XX:YG=YY:REM CO-ORDINATES FOR PLOTTING G
RAPHICS
476 ZZ=(AD-1)*5+AC:ZG=ZZ:REM ** LOCATION IN ARRAY
477 IF ZF=ZG THEN PLOT 29,1,"Silly!":GOTO600
478 IF A(ZZ)=0 THEN PLOT 29,1,"Silly!":KK=1:GOTO600
480 GOSUB6000:REM PLOT GRAPHICS
490 PLOT30,20,1
590 PLOT31,20,"<SPACE>"
600 REPEAT
610 GET FA$
620 UNTIL FA$=" "
630 PLOT30,13,"          "
640 PLOT30,8,"          "
650 PLOT32,15,"  ":PLOT32,10,"  "
655 PLOT29,1,"      ":REM 6 SPACES
660 PLOT 30,20,"         "
680 IFKK<>1 THEN GOSUB 5000:REM ERASE GRAPHICS
790 UNTIL E%=1 OR SC =10 :REM REPEAT UNTIL WON OR RAN OUT OF TURNS
800 IF SC=10 THEN GOTO 7000
810 CLS:FORX=1TO2:PLOT1,X,X:PLOT2,X,14:PLOT10,X,"YOU LOST!!!":NEXT
814 FOR X=1 TO 4:FORY=1TO150:SOUND1,Y,8:SOUND2,255-Y,8:NEXTY,X
815 SOUND1,0,0:SOUND2,0,0
820 PRINT:PRINT:PRINT:PRINT"Want another go   <Y>es or <N>o ?"
830 REPEAT
840 GET G$
850 UNTIL G$="Y" OR G$="N"
860 IF G$="Y"THENPRINTCHR$(17):RUN
870 PRINT:PRINT"GOODBYE... THANK YOU FOR PLAYING"
880 PRINTCHR$(17):REM TURN ON CURSOR AGAIN
890 END
999 REM ** SUBROUTINE TO DRAW BOARD FOR THE GAME **
1000 AA=16:AB=23:AC=2:AD=0
1001 CLS
1002 FOR A = 1 TO 4
1005 FOR X = 1 TO 5
1010 PLOT3,X+AC,AA
1020 PLOT8,X+AC,AB
1030 PLOT13,X+AC,AA
1040 PLOT18,X+AC,AB
1050 PLOT23,X+AC,AA
1060 PLOT28,X+AC,22
1070 NEXT X
1080 F=FRE("")
1090 AD=AA:AA=AB:AB=AD:AC=AC+5
1100 NEXT A
1110 FOR A=1 TO 4
1115 AA$=CHR$(64+A)
1120 PLOT0,A*5,1:PLOT1,A*5,AA$
1125 F=FRE("")
1130 NEXT A
1140 PLOT1,1,1
1145 FOR A = 1 TO 5
1150 AA$=CHR$(48+A)
1160 PLOT 5*A-1,1,AA$
1170 F=FRE("")
1180 NEXT
1190 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
1200 RETURN
1999 REM ** SUBROUTINE TO  FILL ARRAY FOR BOARD LOCATIONS **
2000 CLS:PRINTSPC(7);"Please wait a moment"
2005 FOR X = 1 TO 20
2010 REPEAT
2020 Y= INT(RND(1)*20)+1
2030 IF Y>20 OR Y <1 THEN PRINT"ERROR  in the random number generat
ion":END
2035 REM ** CHECK TO SEE IF ARRAY ELEMENT IS EMPTY OR NOT
2040 UNTIL A(Y)=0
2050 A(Y) = X
2060 NEXT
2070 RETURN
3999 REM ** SUBROUTINE FOR TITLES **
4000 PAPER 0 :INK 6
4005 CLS:FORX=1TO2
4010 PLOT1,X,14:PLOT2,X,X
4020 PLOT12,X,"Monster Hunt"
4030 NEXT
4040 PRINT:PRINT:PRINT:PRINT:PRINT
4050 PRINT"The game is played on a 5 x 4 board,"
4060 PRINT"labelled 1 - 5 across and A to D down."
4070 PRINT"You must tell the computer which"
4080 PRINT"square you wish to view by typing in"
4090 PRINT"the co-ordinates of the square in the"
4100 PRINT"form of number letter (eg. 4D or 2A)"
4101 RESTORE
4102 FOR T=1 TO 34:READ X
4103 MUSIC 1,3,X,0
4104 PLAY 1,0,4,500
4105 WAIT 10
4106 NEXT:PLAY0,0,0,0
4110 PLOT1,22,1:PLOT5,22,"Press <Space Bar> to continue"
4120 REPEAT
4130 GETH$
4140 UNTIL H$=" "
4150 FOR X = 5 TO 13
4160 PLOT1,X,Z$
4170 NEXT
4172 FOR X=5 TO 8:PLOT0,X,7:NEXT X
4180 PLOT1,5,"Hidden on this grid are 20 monsters."
4190 PLOT1,6,"There are 2 different types of "
4200 PLOT1,7,"monster hidden on the board and they"
4210 PLOT1,8,"are found in five different colours."
4220 FOR X = 15 TO 20 :PLOT 0,X,3 :NEXT X
4230 PLOT 1,15,"You have only twenty turns in which "
4240 PLOT1,16,"you must try to discover the location"
4250 PLOT1,17,"of all ten pairs of identical monsters"
4260 REPEAT
4270 GET H$
4280 UNTIL H$=" "
4290 RETURN
5000 DD=A(ZF):DE=A(ZG)
5010 IF(DD=1ANDDE=2)OR(DD=2ANDDE=1)THEN SC=SC+1:A(ZF)=0:A(ZG)=0:RET
URN
5020 IF(DD=3ANDDE=4)OR(DD=4ANDDE=3)THEN SC=SC+1:A(ZF)=0:A(ZG)=0:RET
URN
5030 IF(DD=5ANDDE=6)OR(DD=6ANDDE=5)THEN SC=SC+1:A(ZF)=0:A(ZG)=0:RET
URN
5040 IF(DD=7ANDDE=8)OR(DD=8ANDDE=7)THEN SC=SC+1:A(ZF)=0:A(ZG)=0:RET
URN
5050 IF(DD=9ANDDE=10)OR(DD=10ANDDE=9)THEN SC=SC+1:A(ZF)=0:A(ZG)=0:R
ETURN
5060 IF(DD=11ANDDE=12)OR(DD=12ANDDE=11)THEN SC=SC+1:A(ZF)=0:A(ZG)=0
:RETURN
5070 IF(DD=13ANDDE=14)OR(DD=14ANDDE=13)THEN SC=SC+1:A(ZF)=0:A(ZG)=0
:RETURN
5080 IF(DD=15ANDDE=16)OR(DD=16ANDDE=15)THEN SC=SC+1:A(ZF)=0:A(ZG)=0
:RETURN
5090 IF(DD=17ANDDE=18)OR(DD=18ANDDE=17)THEN SC=SC+1:A(ZF)=0:A(ZG)=0
:RETURN
5100 IF(DD=19ANDDE=20)OR(DD=20ANDDE=19)THEN SC=SC+1:A(ZF)=0:A(ZG)=0

:RETURN

5110 FOR N = 0 TO 2
5120 PLOTXF,YF+N,"   ":REM 3 SPACES
5130 PLOTXG,YG+N,"   ":REM 3 SPACES
5140 NEXT N
5500 RETURN
5998 REM ** ROUTINE FOR PLOTTING GRAPHICS **
5999 REM ** FIRST SELECT COLOUR CO **
6000 L=A(ZZ)
6010 IF (L<5) THEN CO=1
6020 IF (L>4 AND L<9) THEN CO=2
6030 IF (L>8 AND L<13) THEN CO=3
6040 IF (L>12 AND L<17) THEN CO=4
6045 REM ** CO = COLOUR
6050 IF (L>16) THEN CO=5
6060 FOR N=0 TO 2 :PLOTXX,YY+N,CO: NEXT
6080 IF L=1ORL=2ORL=5ORL=6ORL=9ORL=10ORL=13ORL=14ORL=17ORL=18THENL=
0ELSEL=1
6090 REM L=0 TYPE 1 MONSTER L=1 TYPE 2 MONSTER
6100 IFL=0 THENPLOTXX+1,YY,TA$:PLOTXX+1,YY+1,TB$:PLOTXX+1,YY+2,TC$
6110 IFL=1 THENPLOTXX+1,YY,UA$:PLOTXX+1,YY+1,UB$:PLOTXX+1,YY+2,UC$
6120 RETURN
7000 CLS:PRINTSPC(10)"YOU WIN!!!!"
7010 PRINT:PRINT:PRINT:PRINT:PRINT
7020 PRINT"Want another go   <Y>es or <N>o ?"
7021 RESTORE
7022 FOR T=1 TO 34
7023 READX
7024 MUSIC1,3,X,0
7025 PLAY1,0,4,500
7026 WAIT 10
7027 NEXT
7028 PLAY0,0,0,0
7030 REPEAT
7040 GET G$
7050 UNTIL G$="Y" OR G$="N"
7060 IF G$="Y"THENPRINTCHR$(17):RUN
7070 PRINT:PRINT"GOODBYE... THANK YOU FOR PLAYING"
7080 PRINTCHR$(17):REM TURN ON CURSOR AGAIN
7090 DATA8,5,8,5,8,4,8,5,7,5,6,4,7,3,8,6,9,4,9,5,6,3,8,7,6,5,4,7,4,
8,2,9,4,5
7100 END
```

ESCAPE FROM ALPHA CENTURI

Can you take pity on and help an Alpha Centuri inhabitant escape from his disintegrating planet?



Please help me! I have lived on the planet of Alpha Centuri 11 all my life, but the planet has started to disintegrate. Most of the other inhabitants have already escaped and I must try to do the same. There are only 100 minutes left before the planet finally explodes and each move that I make takes one minute! You must give me instructions to help me to escape.

WORDS UNDERSTOOD

N,S,E,W, GO, IN, OUT, UP, DOWN, GET, TAKE, DROP, GRAB, LEAVE, INVENTORY, EAT, DRINK, RUB, SWIM, PADDLE, WADE, UNLOCK, INSERT, WEAR, FILL, JUMP, CLIMB, REPAIR, MEND, FIX, HELP, START, FIRE, WAIT, LAND, PRESS, PUSH, CRAWL, PULL.

I don't speak very good English and if I don't seem to understand you, try another word.

LIST OF VARIABLES

P%	Current location.
S%(X,Y)	Holds map.
B%(X,1)	Holds location items are found in
Q$(X)	Holds descriptions of locations.
G$(X)	Holds descriptions of items.
N$(X)	Holds words understood.
AA-AK	Act as flags to test if you have performed tasks correctly
V$(X)	Holds items carried.
Z$,C$,B$,L$	Hold words entered.
M%	Holds number of moves made.

RUNdown

Lines	Action
50	Selects text mode and chooses colours of foreground/background (can be omited for other machines . . . BBC users choose MODE 7)
55	Dimensions arrays.
60	Releases high resolution memory area (only needed in 16K machines).
65	Turn off cursor.
70	Calls instructions subroutine.
80-110	Prints author's name whilst data is being read.
110-220	Read data into arrays.
225	Selects initial conditions.
230-900	Main control loop (REPEAT UNTIL ran out of moves or won game).

▶

CONVERSION CLUES

1. The program makes extensive use of the REPEAT UNTIL loop and is therefore extremely easy to convert to run on the BBC Micro and Electron, but not quite so easy for other machines.

2. BBC Micro users—the only changes necessary are:

a) PING, ZAP, SHOOT, EXPLODE: replace by other sound effects.

b) Replace music in lines 5080-5095 and 955-962 with tune of your choice . . . see data lines 10000-10010.

c) Use MODE 7 . . . leave CHR$ exactly as they are — they select colours in teletext mode. Electron users should use MODE 6 and delete references to CHR$ (129)-CHR$ (145).

d) PLOT statements should be replaced with PRINT TAB (X,Y) in lines 5160-5240 and 5530-4435 (which plot teletext graphics).

Note:

Some lines may seem to be too long to type into the ORIC1. These lines can be entered if the abbreviation ? is used for PRINT. You do not need to leave a space after the line number, as the computer will automatically insert one for you.

232-350	Decide where you can move from location you are in.
351	Garbage collection . . . not needed on most machines.
360-400	Print items you find in location.
410	Input action.
420	Analyse either first two letters or first three letters of word.
425	Garbage collection and increase number of moves.
430-570	Acts upon words understood.
901-902	RAn out oif time.
915-1000	Won game.
1010-1420	GET item.
1490-1540	Check item is there.
1599-1700	DROP routine.
1740-1780	INVENTORY.
1799-1810	Lose routine.
1819-1840	SWIM.
1849-1865	Paddle routine.
1869-1885	Unlock.
1890-1910	Wear routine.
1920-1946	IN/OUT routines.
1950-1966	Fill routine.
1970-1997	Repair routine.
2000-2010	Help routine.
2020-2510	Start engines.
2520-2540	Land spaceship.
2900-3200	Push/pull/press routines.
5000-5070	Titles.
5080-5095	Play music . . . data for this held in lines 1000-10010.
5110-5130	Wait for space bar to be pressed.
5150-5280	Instructions.
5500-5550	Teletext graphics.
9999-10390	DATA for game.

```
10 REM ** 'ESCAPE FROM ALPHA CENTURI **
20 REM ** AN ADVENTURE GAME FOR THE ORIC 1  of all memory sizes **
30 REM **      S.W. LUCAS   January 1984 **
40 REM ** VERSION 1A **
50 TEXT:PAPER 0:INK 7
55 DIM Q$(30),S%(30,4),X$(40),G$(40).B%(40,1),N$(40),N%(40)
60 GRAB :REM TO RELEASE THE HIGH RESJLUTION MEMORY AREA FOR 16K USE
RS
65 PRINTCHR$(17):REM TURN OFF CURSOR
70 GOSUB 5000:REM INSTRUCTIONS
75 DIM Q$(30),S%(30,4),X$(40),G$(40),B%(40,1),N$(40),N%(40)
80 CLS
90 FORX=1TO2:PLOT1,X,X:PLOT2,X,14
100 PLOT9,X,"    S.W. Lucas  1984"
110 NEXT
120 FOR H=1 TO 28
130 READ Q$(H)
140 FOR D=1 TO 4
150 READ S%(H,D)
160 NEXT D,H
170 FORX=1TO18
180 READG$(X),B%(X,1)
190 NEXT
200 FOR X=1 TO 27
210 READ N$(X),N%(X)
220 NEXT
225 P%=1:M%=0
230 CLS:REPEAT
232 IFAF=1THENS%(9,1)=25ELSES%(9,1)=0
233 IFP%=6ANDAK<>1THENPRINT"I trip and fall in the dark":GOSUB1800
235 PRINTCHR$(131)"There are only ";100-M%;"moves left!"
240 PRINTCHR$(133)"I am :-":PRINTQ$(P%):A$=""
250 IFS%(P%,1)<>0THENA$="North"
255 IFS%(P%,2)<>0AND LEN(A$)>0THENA$=A$+", South"
260 IFS%(P%,2)<>0AND LEN(A$)=0THENA$="South"
265 IFS%(P%,3)<>0AND LEN(A$)>0THENA$=A$+", East"
270 IFS%(P%,3)<>0AND LEN(A$)=0THENA$="East"
275 IFS%(P%,4)<>0AND LEN(A$)>0THENA$=A$+", West"
280 IFS%(P%,4)<>0AND LEN(A$)=0THENA$="West"
290 IFP%=27THENA$="OUT"
345 PRINT:PRINT:PRINT
349 IFA$=""THENA$="I don't know where!"
350 PRINTCHR$(134)"I can go :-":PRINTA$
351 F=FRE("")
360 E=0:FORX=1TO18:PP%=0:IFB%(X,1)=P%THENPP%=1
370 IFPP%=1THEN390
380 NEXT:GOTO410
385 PRINT:PRINT
390 IFE=0THENPRINTCHR$(131)"I can see :-"
400 PRINTG$(X):E=E+1:GOTO380
410 PRINT:PRINT:PRINTCHR$(129)"What should I do now ";:INPUTZ$
415 PING:CLS:PRINTCHR$(130)"Please wait a second"
420 B$=LEFT$(Z$,2):C$=LEFT$(Z$,3)
425 AA=FRE(""):M%=M%+1
430 IF C$="LOO"THENK=1
435 IFC$="SAY"ORC$="TAL"ORC$="ASK"THENPRINT"I'm not talking to myse
lf":K=1
440 IF(B$="N"ORLEFT$(Z$,4)="GO N")ANDS%(P%,1)<>0THENP%=S%(P%,1):K=1
445 IF(B$="S"ORLEFT$(Z$,4)="GO S")ANDS%(P%,2)<>0THENP%=S%(P%,2):K=1
450 IF(B$="E"ORLEFT$(Z$,4)="GO E")ANDS%(P%,3)<>0THENP%=S%(P%,3):K=1
455 IF(B$="W"ORLEFT$(Z$,4)="GO W")ANDS%(P%,4)<>0THENP%=S%(P%,4):K=1
460 IFC$="GET"ORC$="TAK"ORC$="GRA"THENGOSUB1010
465 IFC$="DRO"ORC$="LEA"THENGOSUB1600
470 IFC$="INV"THENGOSUB1750
475 IFC$="EAT"THENPRINTCHR$(131)"There's nothing here to eat!":K=1
480 IFC$="DRI"THENPRINTCHR$(133)"There's nothing to drink!":K=1
485 IFC$="RUB"THENPRINT"Don't be silly!":K=1
490 IFC$="SWI"THENGOSUB1820
495 IFC$="PAD"ORC$="WAD"THENGOSUB1850
500 IF C$="UNL"ORC$="INS"THENGOSUB1870
505 IFC$="SCO"THENPRINT"What do you think this is... a game?":K=1
510 IFC$="WEA"THENGOSUB1890
520 IFC$="IN"ORLEFT$(Z$,4)="GO I"ORLEFT$(Z$,4)="GO L"THENGOSUB1920
525 IFC$="OUT"ORLEFT$(Z$,3)="GO O"THENGOSUB1940
530 IFC$="FIL"THENGOSUB1950ELSEIFC$="JUM"THENGOSUB1970
534 IFC$="CLI"ORC$="UP"THENGOSUB1980
535 IFC$="REP"ORC$="MEN"ORC$="FIX"THENGOSUB1990
540 IFC$="HEL"THENGOSUB2000ELSEIFC$="STA"ORC$="FIR"THENGOSUB2020
545 IFC$="WAI"THENGOSUB2500
550 IFC$="LAN"THENGOSUB2520
560 IFC$="PRE"ORC$="PUS"THENGOSUB2900
565 IFC$="CRA"ORLEFT$(Z$,4)="GO V"THENGOSUB3000
570 IFC$="PUL"THENGOSUB3200
900 UNTIL P%=28 OR M%=100
901 IFM%>99 THENPRINT"YOU RAN OUT OF TIME!":EXPLODE:WAIT100
902 IFM%>99THENSOUND1,0,0:WAIT10:GOTO1800
910 CLS:FORX=1TO2:PLOT1,X,X:PLOT2,X,14
915 PRINT:PRINT:PRINT
920 PLOT10,X,"W E L L  D O N E"
930 NEXT
940 PRINT:PRINT:PRINT:PRINT"You have solved this adventure and"
950 PRINT:PRINT"have escaped from the doomed planet!"
955 RESTORE
956 FOR T=1TO34:READX
957 MUSIC1,3,X,0
958 PLAY1,0,4,500
960 WAIT10
961 NEXT
962 PLAY 0,0,0,0
```

```
3005 IFP%=3THENP%=2:PRINT"O.K.":RETURN
3010 PRINT"DON'T BE DAFT":RETURN
3200 K=1:IFP%<>7THENPRINT"Not here!":RETURN ELSES%(7,2)=8:PRINT"O.K
":RETURN
4999 REM ** INSTRUCTIONS **
5000 GOSUB5500:REM ** PLOT GRAPHICS **
5010 FORX=5TO6:PLOT1,X,X-2:PLOT2,X,14
5020 PLOT7,X,"ESCAPE FROM ALPHA CENTURI"
5030 NEXT
5040 PRINT:PRINT:PRINT:PRINT:PRINT
5050 PLOT4,10,"An adventure game for the ORIC1"
5060 PLOT1,15,3:PLOT17,15,"by"
5070 PLOT1,17,6:PLOT12,17,"S. W. Lucas"
5080 RESTORE
5085 FOR T=1TO34:READX
5086 MUSIC1,3,X,0
5090 PLAY1,0,4,500
5092 WAIT 10
5094 NEXT
5095 PLAY 0,0,0,0
5100 PLOT1,24,5:PLOT6,24,"Press <Space Bar>  to continue"
5110 REPEAT
5120 GETS$
5130 UNTIL S$=" "
5140 GOSUB5500:REM PLOT GRAPHICS
5150 PRINT:PRINT:PRINT:PRINT:PRINT
5160 PLOT10,6,"Please help me !"
5170 PLOT3,10,"I have lived on the planet "
5180 PLOT3,11,"Alpha Centuri all my life, but the "
5190 PLOT3,12,"planet has started to disintegrate."
5200 PLOT3,15,"There are only 100 minutes left "
5210 PLOT3,16,"before the planet finally explodes"
5220 PLOT3,17,"and each move that I make takes "
5230 PLOT3,18,"exactly one minute."
5240 PLOT1,24,5:PLOT6,24,"Press <Space Bar>  to continue"
5250 REPEAT
5260 GETS$
5270 UNTIL S$=" "
~~5280 RETURN~~
5480 END
5499 REM PLOT GRAPHICS
5500 CLS
5510 FOR X=1 TO 2
5520 PLOT0,X,X:PLOT0,X+20,X
5525 PLOT1,X,9:PLOT1,X+20,9
5530 PLOT2,X+20,"GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG"
5535 PLOT2,X,"GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG"
5540 NEXT
5550 RETURN
9592 WAIT 10
9999 REM ** DATA FOR MUSIC
10000 DATA1,1,3,1,1,5,1,1,3,1,5,1,3,5,6,8,1
10010 DATA1,3,5,6,8,1,6,1,8,1,12,10,8,6,5,3,1
10020 REM ** DATA FOR LOCATIONS OF GAME
10030 DATAin a glass walled corridor. There is asupply shop to the
left.
10032 DATA0,0,2,0
10040 DATAin a narrow corridor. A tiny air vent leads to the North.
,0,5,0,1
10050 DATAin a small room full of strange tools.An air vent leads S
outh.
10055 DATA 0,0,4,0
10060 DATAin a storeroom. There is not much here,0,0,0,3
10070 DATAin the entrance to a jet black room.,2,0,6,0
10080 DATAin the jet black room. I can't see a  thing!,0,7,0,5
10090 DATAby a large statue. There is a lever   on the wall.,6,0,0,
0
10100 DATAon a narrow undulating walkway.,7,0,9,10
10110 DATAat the edge of a muddy field. I would need MUCCA BOOTS to
 go North
10115 DATA0,0,0,8
10120 DATAoutside a SPACEX LIBRARY. It is locked,0,11,8,0
10130 DATAby a Kolonian Furitur Fountain.,10,12,13,0
10140 DATAin a small harbour.There is an island only a short distan
ce away.
10145 DATA11,17,14,0
10150 DATAoutside the ALPHA BANK. There is a    cash machine here.
10155 DATA0,14,0,11
10160 DATAoutside the Voltex Souvenir shop. It  is locked.,13,0,15,
12
10170 DATAon a small bridge over the river ZOLPA,0,16,0,14
10180 DATAon the banks of the river. I can see  a small island in t
he river.
10182 DATA15,0,0,0
10185 DATAin the spaceport. The spaceship POLON is here. It is brok
en.
10190 DATA12,0,0,0
10200 DATAon a small island. There is a stand   which holds the key
s.
10205 DATA0,0,0,0
10210 DATAon an island. There is a LOGOR plant  here. The mainland
is to N.
10215 DATA 0,20,0,0
10220 DATAby a scrapyard full of rusty SPACORS.,19,0,0,0
10230 DATAin the cockpit of the spaceship.
10234 DATA0,0,0,0
10235 DATA flying above the clouds. The dial     shows red.,0,0,0,0
10240 DATA in outer space. The planet has just   exploded.,0,0,0,0
10241 DATAin the library. It is full of hologramimages. There are a
 few books
10242 DATA0,10,0,0
10243 DATA on a narrow ledge.,0,9,0,0
10244 DATA in a fuel station. A rope hangs from  above.,0,0,0,0
10245 DATA in a supply shop.,0,0,0,0
~~10250 DATA in the spacestation URIVALINORIC.,0,0,0 0~~
10260 REM ** DATA FOR ITEMS TO BE FOUND
10270 DATAa space helmet,4,a polvar extractor,3,a pile of dust,2
10280 DATAa pressure pad,5,a golden ring,6,a tiny spider,7
10290 DATAa poor beggar,8,a plank,9,a keyhole,10
10300 DATAa toffee paper,11,a sailboard,12,a purse full of coins,13
10310 DATA a bunch of keys,18,some flowers,19,a petrol can,20
10320 DATA a repair manual,24,a pair of MUCCA boots,27
10330 DATAa coin operated rocket fuel pump,26
10340 REM ** WORDS UNDERSTOOD **
10350 DATAHELMET,1,POLVAR,2,EXTRACTOR,2,DUST,3,PRESSURE,4,PAD,4
10360 DATA GOLD,5,RING,5,SPIUNTILER,6,BEGGAR,7,PLANK,8
10370 DATAHOLE,9,TOFFEE,10,PAPER,10,SAILBOARD,11,PURSE,12,COINS,12
10380 DATAKEYS,13,FLOWERS,14,PETROL,15,CAN,15,REPAIR,16,MANUAL,16
10390 DATA MUCCA,17,BOOTS,17,FUEL,18,PUMP,18
```

```
965 PRINT:PRINT:PRINT"Would you like another game <Y/N>"
970 REPEAT
980 GETG$
990 UNTIL G$="Y"ORG$="N"
995 IFG$="Y" THEN PRINTCHR$(17):RUN
1000 END
1009 REM ** GET ROUTINE
1010 K=1:GOSUB1500:REM CHECK ITEM THERE
1020 IFL%<>1THENRETURN
1030 E%=0:FORX=1TO18:IFB%(X,1)=P%ANDB%(N%(R),1)=P%THENE%=1
1040 NEXT:IFE%=0THENPRINTCHR$(129)"I  don't see it here":RETURN
1050 IFR=2ORR=3THENPRINT"AAGGHH...the agony!"
1055 IFR=2ORR=3THENPRINT"I've broken my back with the strain":GOSUB
1800
1060 IFR=1THENAJ=1ELSEIFR=4THENPRINT"The dust blows into my face an
d"
1065 IFR=4THENPRINT"and chokes me to death!":GOSUB1800
1070 IFR=5ORR=6THENPRINT"Don't be silly!":RETURN
1075 IFR=9THENPRINT"AAAAGGGG....It bit me and the poison"
1080 IFR=9THENPRINT"surges through my veins":GOSUB1800
1085 IFR=10ORR=12THENPRINT"Now then be sensible!":RETURN
1090 IFR=16ORR=17THENAB=1ELSEIFR=18THENAC=1ELSEIFR=20ORR=21THENAD=1
1095 IFR=22ORR=23THENAE=1ELSEIFR=25ORR=26THENAF=1
1100 IFR=27THENPRINT"I can't carry a fuel pump that size!":RETURN
1105 IFR=26ANDAD<>1THENPRINT"I need something to put the fuel in!":
~~RETURN~~
1110 IFR=26ANDAD=1THENPRINT"I fill the can with fuel":AG=1
1400 E%=0:FORD=1TO3:IFV$(D)=""THENV$(D)=G$(N%(R)):E%=1:D=5
1410 NEXT:IFE%=0THENPRINT"My hands are full!":K=1:RETURN
1420 B%(N%(R),1)=0:K=1:RETURN
1490 REM ** CHECK ITEM IS THERE
1500 L$="":FORH=1TOLEN(Z$)
1510 IFMID$(Z$,H,1)=" "THENL$=RIGHT$(Z$,(LEN(Z$)-H)):H=H+40
1520 NEXT:R=0:L%=0:IFLEN(L$)<2THENK=1:RETURN
1530 FORH=1TO27:IFLEFT$(N$(H),LEN(L$))=L$THENL%=1:R=H
1540 NEXT:K=1:RETURN
1599 REM ** DROP ITEM
1600 K=1:GOSUB1500:IFL%<>1THENPRINT"I cant see a :-";L$:RETURN
1605 E%=0:FORD=1TO3:IFV$(D)=G$(N%(R))THENV$(D)="":E%=1
1610 NEXT:IFE%<>1THENPRINT"I haven't got it!!":RETURN
1615 B%(N%(R),1)=P%
1620 IFR=1THENAJ=0 ELSEIFR=16ORR=17THENAB=0 ELSEIFR=18THENAC=0
1625 IFR=20ORR=21THENAD=0 ELSEIFR=22ORR=23THENAE=0
1630 IFR=24ORR=25THENAF=0
1635 IFAG=1AND(R=20ORR=21)THENAG=0:PRINT"The fuel spills to the flo
or"
1700 RETURN
1740 REM ** INVENTORY
1750 PRINTCHR$(129)"I  have :-":F%=0:FORH=1TO3
1760 IFV$(H)<>""THENPRINTV$(H):F%=1
1770 NEXT:IFF%=0THENPRINT"Nothing at all"
1780 PRINT:PRINT:K=1:RETURN
1799 REM DEATH ROUTINE
1800 PRINT:PRINT:PRINT:PRINT"With friends like you who needs   "
1810 PRINT"enemies ?":PRINT:PRINT:PRINT:GOTO955
1819 REM ** SWIM **
1820 K=1:IFP%=18THENP%=12:PRINT"O.K":RETURN
1825 IFP%=12THENP%=18:PRINT"O.K.":RETURN
1830 IFP%=16ORP%=19THENPRINT"The water isn't deep enough!":RETURN
1840 PRINT"Don't be stupid!":RETURN
1849 REM ** PADDLE *
1850 K=1:IFP%=12ORP%=18THENPRINT"The water's too deep... I drown":P
RINT:PRINT
1855 IFP%=12ORP%=18THENGOSUB1800
1860 IFP%=16THENPRINT"O.K.":P%=19:RETURN
1862 IFP%=19THENP%=16:PRINT"O.K.":RETURN
1865 PRINT"Don't be stupid!!!":RETURN
1869 REM UNLOCK
1870 IFP%<>10THENPRINT"Why do you always have to be so silly?":K=1:
RETURN
1875 K=1:IFAC<>1THENPRINT"I don't have the key!":RETURN
1880 PRINT"O.K.":S%(10,1)=24:Q$(10)=LEFT$(Q$(10),25):XX=FRE("")
1885 RETURN
1890 IFAJ=1THENPRINT"I am already wearing the helmet"
1900 IFAF=1THENPRINT"I've already put them on!"
1905 K=1:IFAF=0ANDAA=0THENPRINT"Don't be silly"
1910 RETURN
1920 K=1:IFP%=1THENP%=27:PRINT"O.K.":RETURN
1922 IFP%=17THENP%=21:PRINT"O.K.":RETURN
1930 PRINT"SILLY":RETURN
1940 CLS:IFP%=27THENP%=1:PRINT"O.K."ELSEIFP%=18THENP%=17:RETURN
1945 IFP%=21THENP%=17:PRINT"O.K.":RETURN
1946 RETURN
1950 K=1:IFP%<>17THENPRINT"I can't fill the rocket yet!":RETURN
1960 IFAG<>1THENPRINT"I HAVE NO FUEL":RETURN
1965 PRINT"O.K. I  fill the fuel tank":AH=1
1966 RETURN
1970 IFP%<>25THENPRINT"Not here!":K=1:RETURN
1975 K=1:P%=26:PRINT"O.K.":RETURN
1980 K=1:IFP%<>26THENPRINT"don't be absurd!":RETURN
1985 P%=25:PRINT"O.K.":RETURN
~~1990 K=1:IFP%<>17THENPRINT"I can't see anything in need of repair":~~
RETURN
1994 IFAE<>1THENPRINT"I need the repair manual!":RETURN
1995 PRINT"O.K. I  repair the spaceship!"
1996 Q$(17)=LEFT$(Q$(17),46):AI=1
1997 RETURN
2000 K=1:IFP%=17THENPRINT"WHY NOT REPAIR IT!":RETURN
2005 IFP%=21THENPRINT"WHY NOT START THE ENGINE?":RETURN
2006 IFP%=22THENPRINT"all you can do is wait!":RETURN
2007 IFP%=23THENPRINT"Why not land on that planet I see overthere?"
:RETURN
2010 PRINT"I'm sorry I don't have a clue!":RETURN
2020 IFP%<>21THENPRINT"Not yet":K=1:RETURN
2025 K=1:IFAI<>1THENPRINT"The ship's broken... remember!":RETURN
2030 IFAH<>1THENPRINT"Out of fuel!":RETURN
2035 IFAC<>1THENPRINT"I have no keys!":RETURN
2040 IFAJ<>1THENPRINT"I have no space helmet!":RETURN
2045 P%=22:PRINT"The engine fires!":EXPLODE:WAIT10:EXPLODE:RETURN
2500 K=1:PRINT"O.K.":WAIT100:IFP%=22THENP%=23
2510 RETURN
2520 K=1:IFP%<>23THENPRINT"Don't be stupid!":RETURN
2530 PRINT"O.K. I land the spaceship safely!":P%=28
2540 RETURN
2900 K=1:IFP%=7THENS%(7,2)=8:PRINT"SOMETHING HAPPENED":RETURN
2905 IFP%=5THENPRINT"SOMETHING HAPPENED":AK=1
2906 IFP%=5THENQ$(6)=LEFT$(Q$(6),22):RETURN
2907 PRINT"DON'T BE SILLY":RETURN
3000 K=1:IFP%=2THENP%=3:PRINT"O.K.":RETURN
```

D T Wright

WUMPUS HUNT

Getting into the same square as a Wumpus may sound easy, but you just try it!

With 400 locations to explore, countless enemies to attack or avoid, 10 skill levels and hoards of golden coins to collect; Wumpus Hunt for the 48K Oric is sure to interest many users.

Graphically the game is neat but not really stunning (what d'ya expect with only three user defined graphic characters?). A 20 by 20 grid on the upper left-hand side takes up most of the screen. The three characters which are used signify your present location, locations already visited and unexplored locations. Along two sides of the grid is a scale, this enables the user to note hazardous locations to avoid or locations with items of possible use later on. The present location description and any action is displayed along the bottom 4 lines. And on the right-hand side are the important status numbers. They tell you your strength, weight and energy also the number of gold coins and objects you are carrying.

The aim of the game is to get in the same square as the Wumpus (not as easy as it sounds) and a battle is fought; the outcome depending on your strength (preferably high), your weight (preferably low), your energy (preferably high) and the amount of weapons you're carrying (preferably high). Wumpus' sleep inducer also adds considerably to your chances of capturing the Wumpus. Then you must take it to the teleport station.

CONTROLS

Q	To go North-West.
A	To go West.
Z	To go South-West.
W	To go North.
S	To Fight.
X	To go South.
E	To go North-Esst.
D	To go East.
C	To go South-East.
I	To look at Inventory.
J	To Drop something.
O	To Open box (with Key).
K	To use Strength
P	Toi Quit.
L	To Get something.

THE GAME

The level of difficulty determines the number of friendly and unfriendly people. Only friendly people will sell items. An unfriendly person can (and usually will) attack you. Bats are harmless but super-bats certainly aren't. Zombies wander around the place stealing objects and being unhelpful if possible. Giant spiders can, if in the mood, entangle you in a web.

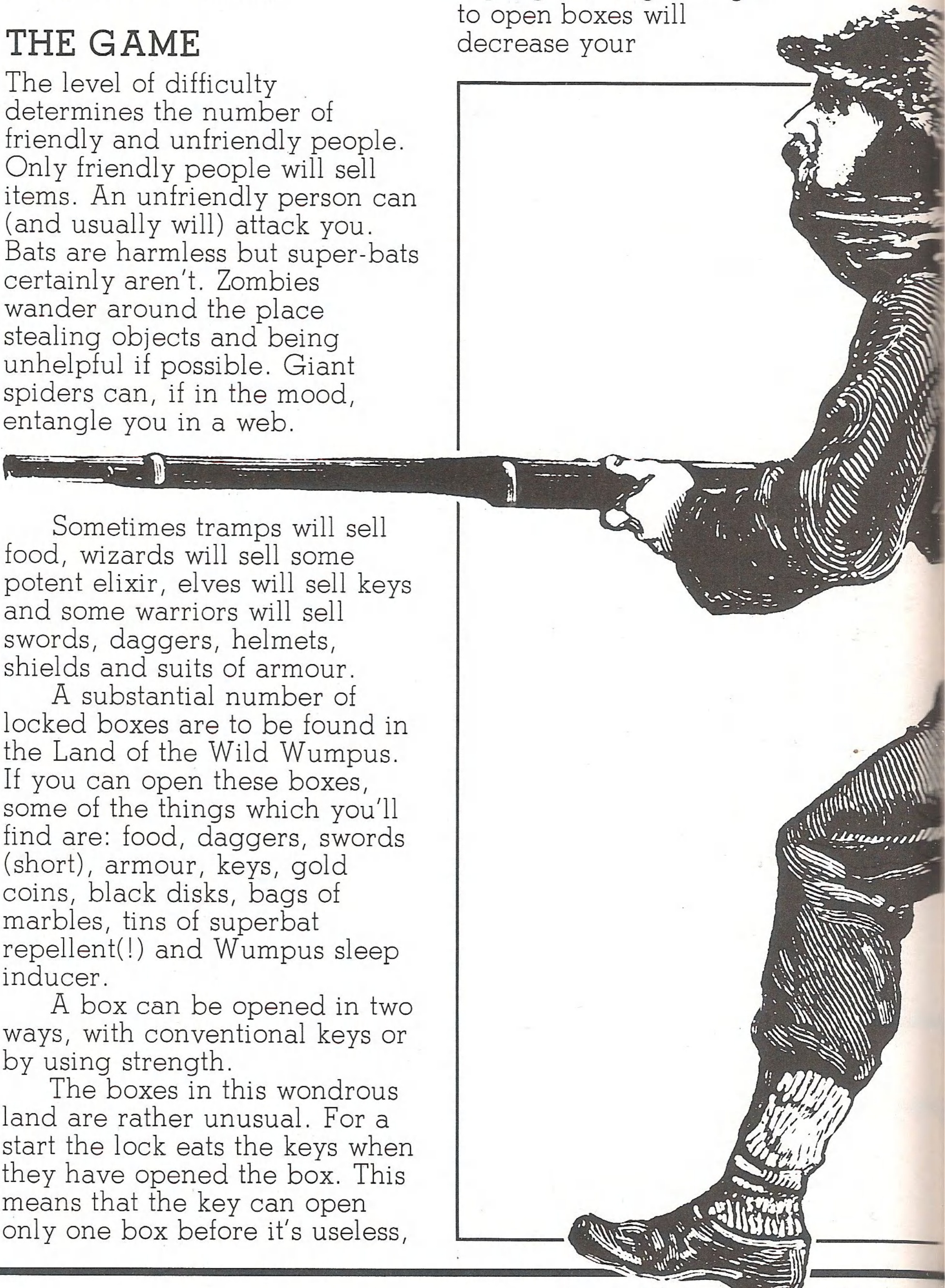
Sometimes tramps will sell food, wizards will sell some potent elixir, elves will sell keys and some warriors will sell swords, daggers, helmets, shields and suits of armour.

A substantial number of locked boxes are to be found in the Land of the Wild Wumpus. If you can open these boxes, some of the things which you'll find are: food, daggers, swords (short), armour, keys, gold coins, black disks, bags of marbles, tins of superbat repellent(!) and Wumpus sleep inducer.

A box can be opened in two ways, with conventional keys or by using strength.

The boxes in this wondrous land are rather unusual. For a start the lock eats the keys when they have opened the box. This means that the key can open only one box before it's useless, in way of recompense the locks 'll eat any ol' key, so every key will open any box.

Also the boxes are sensitive to extreme pressures. If your total energy and strength is too high the box will positively refuse to open and will even threaten to blow the square sky high. Using strength to open boxes will decrease your



strength by 1 each time, the boxes won't always spring open either.

Also featuring in this game of course and cunning are several neutral locations. Swampy, rocky and hilly grounds are common. Fans of 'A Country Practice' will recognise a fat little Wombat which crawls about.

After a few turns you may have noticed that your energy points have decreased. No, it isn't a bug, your Wumpus hunting warrior is probably getting hungry! To increase your energy points, you use the 'get' command on food, or buy it from somebody.

SCORING

For every person killed, you get between 20 and 29 points. Each item that you carry is worth a certain number of points between 2 and 30. The number of gold coins in your possession adds that number to your score. If you manage to capture the Wumpus, you get 1000 points for each man left.

VARIABLES USED.

A1 (19,19,0) (19,19,1)	Holds values for different locations. Values 0 to 33. Values 0 or 1.
B% (19,19,9)	Number of objects in location. Last number corresponds to inventory number. Values 0 to various.
C$ (33)	Descriptions of locations.
D$ (33)	Descriptions of locations.
S,S$	Strength value, in numeric and string form.
W,W$	Weigh values, in numeric and string form.
F,F$	Energy value, in numeric and string form.
O,O$	Number of objects, in numeric and string form.
G,G$	Number of gold coins, in numeric and string forms.
L%	Number of lives left.
P% (6)	Objects which you carry, in numeric form.
P$ (6)	Objects which you carry, in string form.
P1% (6)	Total number of each kind of object possible to carry.
P1$ (6)	Description of each object for inventory.
P2% (9)	Value of each item for score.
K$ (9)	Descriptions of objects for search.
W1 (9)	Weight of objects, for adding to initial weight (or deducting).
L1$(9)	Words for fights.
E$ (9)	Objects in locked boxes, string form (descriptions).
X%,Y%	Co-ordinates for plotting your present location.
Q%,R%	Co-ordinates for plotting locations already visited.
H,H$	High score, in numeric and string form.
N$	Name of high score holder.
B	Skill level input.
I$	General input, as the main input it is in a I$=KEY$ form.
E%	Timer on input.
U2	Enemy advantage when you attack friendly people.
U1	Your advantage if you attack first, increases if you attack superbats with superbat repellent.
U%	If the Wumpus is caught, this has a value of 1, else it has a value of 0.
K%	Numerical value of location, much shorter than the equivalent A1 (X%—2, Y%—1,0).
S%	Points for killings count, added to other counts for score.
Y1	Escaping from grizzly, once caught you don't get caught again there.
Y2	Escaping from giant spiders web, so that once escaped from web you don't get caught in that one again.
C1%	NUmber of escaping grizzly or web.
C2	Number for escaping grizzly or web.
E1	Weight of object for buying routine.
D3%,D2%, C1%	Numbers for calculating price of objects sold.
KA,KC	The two numbers that determine if you have captured the Wumpus when you attack it.

HINTS TO PLAY

It may seem impossible to win the battles at first, the idea is to get some weapons then go off on a killing spree. Don't just go around collecting gold coins either, you're not there to steal the country's money. Anyway, it will just increase your weight until you can't get Wumpus, the Wumpus is your real target.

When you arrive in a square occupied by a hostile person or creature, quickly press the attack key or exit the square. Pressing the attack key before they attack gives you an advantage. Attacking a friendly person gives them an advantage and if you don't kill them, they turn into unfriendly people.

Every object in the game has a purpose. Think about it and you could get a welcome surprise.

CONVERSION CLUES

This program should be fairly easy to convert to other micros, most of it is written in near standard BASIC. Non specific REMs split up the sections or routines.

The screen size of the ORIC is 27 rows by 38 columns. The command PLOT X,Y,Z, is used for plotting. Where X is the column (values 1 to 38), Y is the row (values 0 to 26) and Z is a string (Z$) or an attribute. When the value of Z is between 0 and 7 the attribute alters the foreground colour from the X,Y co-ordinates to the end of the line. When the value of Z is between 8 and 15, the character set used (single height, double height, standard set, alternative set, flashing or steady) is changed from the X,Y co-ordinate to the end of the line. When the value of the attribute is between 16 and 23, the background colour from the X,Y co-ordinate to the end of the line is altered.
PLOT 1,0 "Hi" produces Hi in the top left corner.
XX=FRE("") forces variable garbage collection.
FOR T=48000 TO 48039: POKE T,22:
NEXT pokes a line of cyan background colour along the top of the screen.

RUNdown

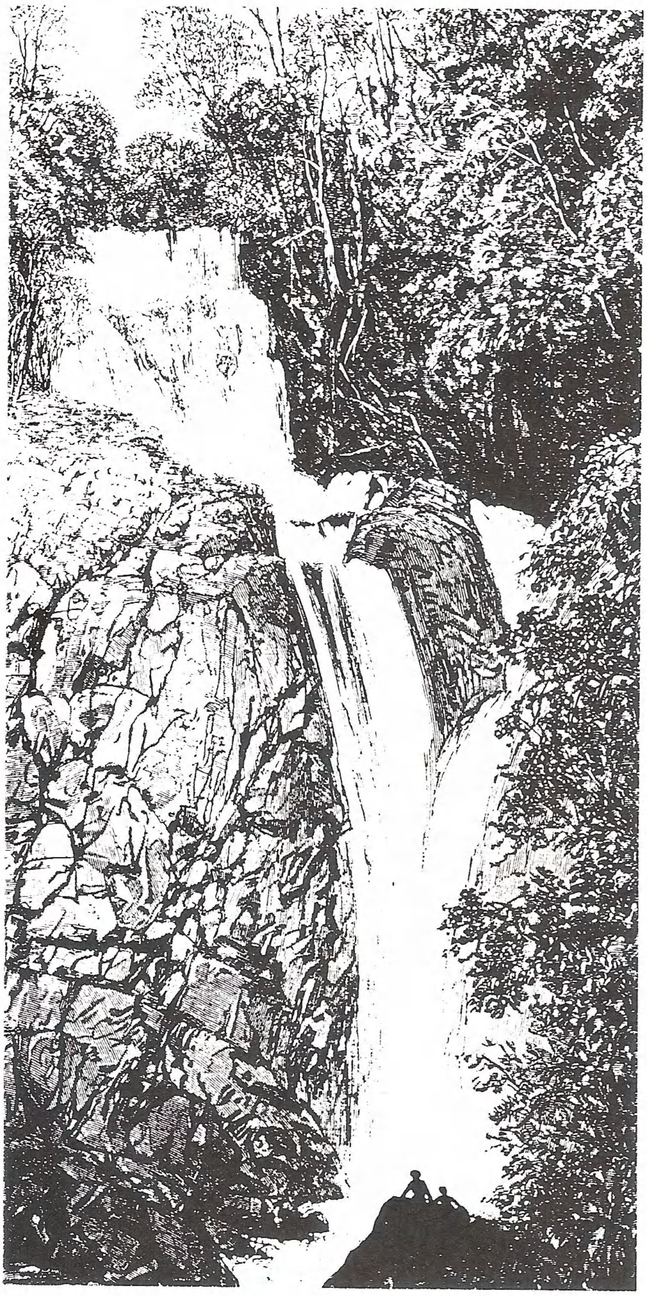
Line	**Action**
1-16	DIMensioning arrays, initialisation, GOTO 1500.
19-44	Main checking routines, updates display, GOTO 110.
70-100	Setting up display routine, used once per game.
110-138	Update location description, jumps to various sections.
144-148	Secondary checking routine.
150-192	Input section, inputs checked and processed.
200-202	Grizzly bear got you routine, jump to escape routine.
210-212	Caught in spiders web routine.
220-244	Bottomless pit routine.
246-248	Life lost routine.
260	Tramp sell data.
270-288	Warrior sell data.
300-302	Wizard sell data.
310-312	Elves sell data.
320-348	Sell routine for sell data.
360-385	Fighting routine, jump to life lost routine.
390-422	Wumpus capture routine.
450-456	Inventory routine.
470-504	Drop object routine.
508-546	Get object routine.
570-592	Open with key routine (box).
620-634	Open with strength routine (box).
700-706	Quit routine.
800-824	Game over routine, jump to little page routine.
850-880	Return to teleport station with Wumpus routine.
1000-1002	Clear bottom 4 lines routine. Most routines jump here.
1500-1554	Initialisation of descriptions of location.
1560-1570	Initialisation of descriptions of contents of boxes.
1580-1590	Redefining characters for display.
1600-1620	Initialisation of various values for objects.
1640-1642	Initialisation of words used in fighting routine.
1700-1720	Title page and tune.
1800-1816	Input skill level, reset variables.
1830-1885	Loops for locating different locations.
2000-2016	Locating different locations.
2100-2122	Putting objects in locations routine.

If you find that the fighting routine is too easy or hard, the IF...THEN statement in line 376 should be altered.

Lines 402 and 404 control the Wumpus capture routine. Line 627 controls the chance of a box exploding. Every time you are required to input something, a PING is heard. If this system proves too noisey, the relevant lines can easily be altered.

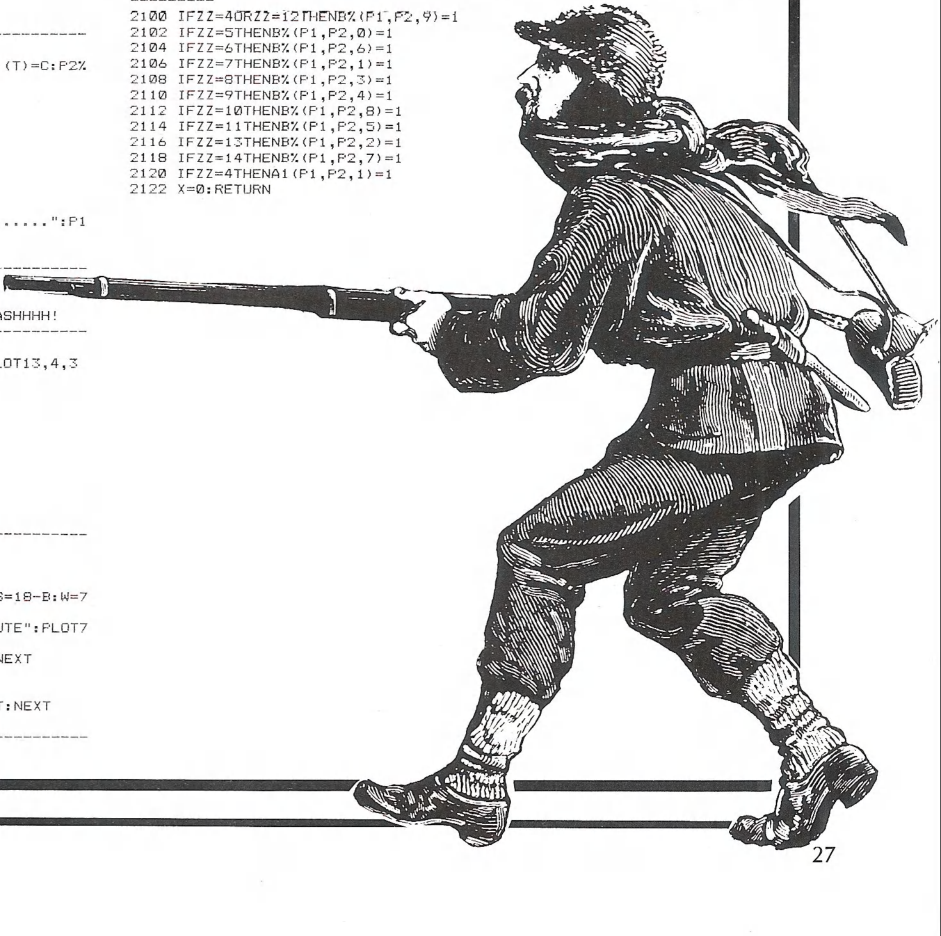
```
1 REM -------------------------------------
2 REM ¦                                   ¦
3 REM ¦                                   ¦
4 REM ¦      W U M P U S   H U N T        ¦
5 REM ¦                                   ¦
6 REM ¦         by  D.T.Wright            ¦
7 REM ¦                                   ¦
8 REM ¦                                   ¦
9 REM -------------------------------------
10 PRINTCHR$(17)CHR$(6):POKE48035,0
12 DIMA1(19,19,1),B%(19,19,9),C$(33),D$(33)
14 N$="First game":H$="000":INK2:PAPER0
16 GOTO1500
18 REM--------------------------------------------------------------
----------
19 Y1=0:Y2=0
20 F=F-.15
21 WAIT60:IFF<4THENS=S-.2
22 IFW>18THENS=S-.2
23 IFS<0THEN246
24 G=INT(G)
25 IFF>16THENS=S+1
26 IFS>20THENS=20
27 IFF>20THENF=20
28 IFF<0THEN246
30 G$=STR$(G):W$=STR$(W):S$=STR$(S):O$=STR$(O):F$=STR$(F)
31 GOSUB1000
32 IFY%>20THENY%=20
33 IFY%<1THENY%=1
34 IFX%>21THENX%=21
35 IFX%<2THENX%=2
36 K%=A1(X%-2,Y%-1,0):I$=""
38 FORT=0TO7:P$(T)=STR$(P%(T)):NEXT
40 FORT=8TO12:PLOT34,T,"     ":NEXT
42 W$=LEFT$(W$,3):S$=LEFT$(S$,3):O$=LEFT$(O$,3):F$=LEFT$(F$,3)
44 GOTO110
69 REM--------------------------------------------------------------
----------
70 CLS:EXPLODE:FORT=48002TO48039:POKET,22:NEXT
72 PLOT1,21,22:PLOT3,21,0:PLOT14,21,"WUMPUS HUNT"
74 FORT=1TO3:PLOT24,T,6:NEXT
76 FORT=0TO14:IFT=5THENNEXT
78 IFT=4ORT=6THENPLOT23,T,22:NEXT
80 IFT=14THENPLOT23,14,22:GOTO84
82 PLOT23,T,22:PLOT24,T,16:PLOT38,T,22:NEXT
84 PLOT26,1,"High-":PLOT30,2,"by":PLOT31,1,H$:PLOTINT(25+((12-LEN(N
$))/2)),3,N$
86 PLOT25,8,"Gold    -":PLOT25,9,"Strength-":PLOT25,10,"Weight  -
88 PLOT25,11,"Objects -":PLOT25,12,"Energy  -":PLOT24,15,"KEY CONTR
OLS
90 PLOT23,16,"W=North P=QUIT":PLOT23,17,"X=South O=Open
92 PLOT23,18,"D=East  L=Get":PLOT23,19,"A=West  I=Invent
94 PLOT23,20,"S=Fight J=Drop":FORT=15TO20:PLOT22,T,5:NEXT
96 FORT=2TO21:FORT2=1TO20:PLOTT,T2,"%":NEXT:NEXT
98 FORT=0TO9:PLOT1,T+1,48+T:NEXT:FORT=0TO9:PLOT1,T+11,48+T:NEXT
100 FORT=2TO21:PLOTT,0,65+(T-2):NEXT:K%=28
109 REM-------------------------------------------------------------
----------
110 PLOTINT((38-LEN(C$(K%)))/2),23,C$(K%):PLOT33,8,G$:PLOT34,9,S$:P
LOT34,10,W$
112 PLOTINT((38-LEN(D$(K%)))/2),24,D$(K%):PLOT34,11,O$:PLOT34,12,F$

114 PLOTQ%,R%,"+":PLOTX%,Y%,"$":E%=0
116 Z=INT(RND(1)*5+B):IFZ<B+2THEN122
118 IFK%=15THEN200
120 IFK%=21THEN210
122 IFK%=23THEN390
124 IFK%=19THEN240
126 Z=INT(RND(1)*5+B):IFZ<B+3THEN136
128 IFK%=2THEN270
130 IFK%=24THEN300
132 IFK%=25THEN310
134 IFK%=3THEN260
136 IFK%=28THEN850
138 GOTO150
144 WAIT200:GOSUB1000
145 FORT=8TO12:PLOT34,T,"    ":NEXT:E%=0
146 PLOTINT((38-LEN(C$(K%)))/2),23,C$(K%):PLOT33,8,G$:PLOT34,9,S$:P
LOT34,10,W$
148 PLOTINT((38-LEN(D$(K%)))/2),24,D$(K%):PLOT34,11,O$:PLOT34,12,F$

150 U1=0:U2=0:I$=KEY$:IFI$=""THENE%=E%+1ELSE160
152 IFE%>30ANDK%=16THEN360
154 IFE%>18ANDK%=18THEN360
156 IFE%>25ANDK%=20THEN360
157 IFE%>1250THENEXPLODE:PLOT12,26,"GET A MOVE ON!":WAIT200:E%=0:GO
TO150
158 IFE%>35ANDK%=21THEN360ELSE150
160 Q%=X%:R%=Y%:I=0:XX=FRE("")
162 IFI$="W"ORI$="w"THENY%=Y%-1:GOTO19
164 IFI$="X"ORI$="x"THENY%=Y%+1:GOTO19
166 IFI$="D"ORI$="d"THENX%=X%+1:GOTO19
168 IFI$="A"ORI$="a"THENX%=X%-1:GOTO19
170 IFI$="Q"ORI$="q"THENX%=X%-1:Y%=Y%-1:GOTO19
172 IFI$="E"ORI$="e"THENX%=X%+1:Y%=Y%-1:GOTO19
174 IFI$="Z"ORI$="z"THENX%=X%-1:Y%=Y%+1:GOTO19
176 IFI$="C"ORI$="c"THENX%=X%+1:Y%=Y%+1:GOTO19
178 IFI$="I"ORI$="i"THEN450
180 IFI$="J"ORI$="j"THEN470
182 IFI$="S"ORI$="s"THENU1=1:GOTO361
184 IFI$="O"ORI$="o"THEN570
186 IFI$="L"ORI$="l"THEN508
188 IFI$="K"ORI$="k"THEN620
190 IFI$="P"ORI$="p"THEN700
192 PING:GOTO144
199 REM-------------------------------------------------------------
----------
200 IFY1=1THEN144
201 X=RND(1)*5:IFX>3.5THEN202ELSE144
202 PLOT1,25,"    You really are in a mess now!    ":C1%=5:C2=2:C3=
2:GOTO220
209 REM-------------------------------------------------------------
----------
210 IFY2=1THEN144
211 X=RND(1)*5:IFX>4.3THEN212ELSE144
212 PLOT1,25," He spins a massive web around you.  ":C1%=6:C2=3:C3=
2
219 REM-------------------------------------------------------------
----------
220 IFS>INT(RND(1)*C1%+B)THEN232
224 WAIT150:GOSUB1000
225 PLOT1,23," SELECT:strength points to escape.":PING:GETI$:I=VAL(
I$)
226 IFI<C1%ANDI<STHENS=S-I:F=F-1:GOTO229
227 IFS<=2THEN246ELSEIFS<ITHENPLOT14,26,"Too high!
228 GOTO224
229 IFI>INT(RND(1)*C2+C3)THEN232
230 PLOT3,26,"Hard cheese!You're still caught.":GOTO224
232 WAIT140:PLOT8,26,"Well done,you've escaped":WAIT150
233 IFC2=3THENY2=1ELSEY1=1
234 GOTO20
239 REM-------------------------------------------------------------
----------
240 WAIT100:GOSUB1000
241 IFS>RND(1)*15+BTHENPLOT4,23,"Because of your strength you."ELSE
245
242 PLOT2,24,"grabbed hold of a branch and so a
244 PLOT8,25,"life hasn't been lost.":S=S-INT(RND(1)*3):GOTO144
245 A1(X%-2,Y%-1,0)=29
246 WAIT50:PLOT1,26,"...HARD CHEDDAR!YOU'VE LOST A LIFE...":L%=L%-1
:S=18-B
248 F=18-B:IFL%=0THEN800ELSEWAIT160:U1=0:GOTO20
259 REM-------------------------------------------------------------
----------
260 PLOT1,25,"...The tramp offers you some food...":D2%=60:D3%=10:C
1%=9:GOTO320
269 REM-------------------------------------------------------------
----------

270 Z=INT(RND(1)*4):IFZ=0THENE1=1:C1%=0:D2%=30:D3%=28:GOTO280
272 IFZ=1THENE1=1:D2%=40:D3%=25:C1%=2:GOTO282
274 IFZ=2THENE1=.4:D2%=50:D3%=15:C1%=1:GOTO284
276 IFZ=3THENE1=3:D2%=38:D3%=70:C1%=2:GOTO286
278 IFZ=4THENE1=1:D2%=45:D3%=26:C1%=2:GOTO288
280 PLOT1,26,"...They offer a fine golden swaord...":GOTO320
282 PLOT1,26,"...A small round shield is offered...":GOTO320
284 PLOT1,26,".They offer a Stay-Sharp (TM) dagger.":GOTO320
286 PLOT1,26,"A suit of royal chain mail is offered":GOTO320
288 PLOT1,26,".They toss a slightly dented helmet..":GOTO320
299 REM-------------------------------------------------------------
----------
300 PLOT1,25,"...The Wizard offers you an elixer...":D2%=20:D3%=100

302 PLOT1,26,"Which increases all of your powers...":C1%=10:GOTO320
309 REM-------------------------------------------------------------
----------
310 PLOT1,26,".....They have many keys to sell.....":D2%=50:D3%=11
312 C1%=3:EL=.15
319 REM-------------------------------------------------------------
----------
320 WAIT200:GOSUB1000
321 I=(INT((RND(1)*10)*(G/D2%))+D3%)
322 PLOT5,26,"and so asks for    gold coins":PLOT17,26,STR$(I)
324 WAIT60:PING:PLOT1,25,"SELECT : (1) yes  (A) Forget it buddy.":G
ETL$
325 PLOT1,26,"                                    "
326 IFL$="A"THEN144
328 IFL$="1"THEN330ELSE324
330 IFG-I<0THENPLOT7,26,"You can't afford it.":GOTO144
332 IFC1%=9THEN342
334 IFC1%=10THEN346
336 IFP%(C1%)=P1%(C1%)THENPLOT6,26,"You can't carry any more.":GOTO
144
338 IFW+E1>20THENPLOT6,26,"You can't carry any more.":GOTO144
339 IFD3%=70THENP%(C1%)=P%(C1%)+3:W=W+E1:O=O+3:G=G-I:GOTO20
340 W=W+E1:O=O+1:P%(C1%)=P%(C1%)+1:G=G-I:GOTO20
342 IFF>18THENPLOT3,26,"Greedy pig!You've already eaten.":GOTO144
344 F=F+2:G=G-I:S=S+.5:GOTO544
346 FORT=1TO12:MUSIC1,3,T,0:PLAY1,0,1,1000:WAIT30-T:NEXT:PLAY0,0,0,
0
348 G=G-I:S=S+6:F=F+6:W=W-4:GOTO20
359 REM-------------------------------------------------------------
----------
360 PLOT1,22,"Unfortunately you have been attacked."
361 IFK%=25THENPLOT1,26,"ELF BASHER!They're an endangered lot!":S=S
-1:GOTO144
362 IFK%=24THENPLOT1,25,"The Wizard roars.Do you dare to attack"ELS
E364
363 PLOT1,26,"me?With one wave,he destroys you.....":WAIT300:EXPLOD
E:GOTO246
364 IFK%=0ORK%=1ORK%=4ORK%=15ORK%=22ORK%=26ORK%=27ORK%=28ORK%=29ORK
%=31THEN369
365 IFK%=30THENPLOT7,26,"STOP TRYING TO RUB IT IN":GOTO144ELSEIFK%=
33THEN369
366 FORT=5TO14:IFK%=TTHENPLOT10,26,"I HOPE YOU'RE WELL":GOTO144ELSE
NEXT
367 IFK%=2ORK%=3ORK%=32THENPLOT10,26,"THAT WASN'T WISE":A1(X%-2,Y%-
1,0)=1~:U2=1
368 GOTO370
369 PLOT6,26,"NOTHING TO ATTACK ERROR.":GOTO144
370 WAIT100:GOSUB1000:PLOT4,23,"SELECT:strength points to attack":P
ING:GETI$
371 I=VAL(I$):IFS<1THEN246ELSEIFI<SANDI<5ANDI>0THENS=S-I:GOTO372ELS
E370
372 WAIT60:GOSUB1000:FORT=1TO6:Z1=INT(RND(1)*9):Z2=INT(RND(1)*25+5)
373 Z3=INT(RND(1)*4+22):PLOTZ2,Z3,L1$(Z1):MUSIC1,0,T,0
374 PLAY1,0,4,1000:WAIT60-T*3:PLOTZ2,Z3,"              ":NEXT:PLAY0,0,0,
0
375 IFK%=18ANDP%(5)>0THENU1=U1+1
376 IFI+P%(0)+P%(1)/2+P%(2)+(16-W)+U1>RND(1)*10+6+U2THEN381
377 PLOT2,24,"Unfortunately you have been beaten.":I=INT(RND(1)*S+F
) :WAIT100
378 IFI<15THENPLOT8,25,"In fact you're now dead..":WAIT60:GOTO246
379 I=INT(RND(1)*4):PLOT3,26,"You have lost   strength points.
380 PLOT16,26,STR$(I):S=S-I:U1=0:WAIT99:GOTO20
381 PLOT1,25,"You are victorious!The foe are beaten":I=INT(RND(1)*2
)
382 L=INT(RND(1)*10+2):S=S-I:PLOT3,26,"You have lost   strength poi
nts.
383 PLOT16,26,STR$(I):WAIT200:PLOT3,26,"You have found    gold coin
s....
384 PLOT17,26,STR$(L):G=G+L:A1(X%-2,Y%-1,1)=1:A1(X%-2,Y%-1,0)=30:WA
IT99
385 O=O+1:U1=0:S%=S%+20+B:GOTO20
389 REM-------------------------------------------------------------
----------
390 WAIT60:PLOT1,23,"     QUICK!        PRESS A KEY         ":E%=0
392 I$=KEY$:IFI$=""THENE%=E%+1ELSE396
```

```
394 IFE%=20THEN414ELSE392
396 PLOT1,24,"SELECT : strength points to attack.":PING:GETI$:ZAP:W
AIT100
398 I=VAL(I$):IFI<5ANDI>0THEN399ELSE396
399 IFS<2THEN246
400 IFI>STHEN396
402 KA=I+P%(0)+P%(1)/2+P%(2)+P%(7)*2+(16-W)+RND(1)*3
404 KC=RND(1)*5+RND(1)*5+10:S=S-I
406 IFKA<KCTHEN414
408 PLOT4,25,"You have captured the Wumpus!":WAIT60
410 PLOT1,26,"Now you must take it to civilisation":U%=1
412 A1(X%-2,Y%-1,0)=32:GOTO144
414 PLOT7,25,"The Wumpus has escaped!
416 PLOT6,26,"And has stolen some gold!":G=INT(G-((G/30)*B+1))
418 X=INT(RND(1)*19):Y=INT(RND(1)*19)
420 IFA1(X,Y,0)<>28THENA1(X,Y,0)=23:A1(X%-2,Y%-1,0)=32ELSE418
422 WAIT150:GOTO20
449 REM--------------------------------------------------------------
----------
450 GOSUB1000:IFK%=17ORK%=16THENPLOT7,26,"You are told to sod off!"
:GOTO144
451 X=RND(1)*3:IFX<1THENPLOT3,25,"Unfortunately a ZOMBIE chases you
"ELSE453
452 PLOT7,26,"and you have to move away":GOTO144
453 PLOT14,22,"INVENTORY":FORT=0TO7:PLOT6,24,P1$(T)
454 PLOT1,24,STR$(T):PLOT3,24,")"
455 PLOT27,24,P$(T):PING:PLOT12,26,"PRESS ANY KEY
456 GETI$:NEXT:GOTO144
469 REM--------------------------------------------------------------
----------
470 WAIT150:GOSUB1000:IFK%=17THENPLOT5,26,"The tramp doesn´t like y
ou":GOTO144
471 PLOT13,22,"DROP OBJECT":PLOT1,24,"SELECT  : object number (A)fo
rget":PING
```



```
472 GETI$:PLOT13,26,"Selected":PLOT23,26,I$:IFI$="A"ORI$="a"THEN144
ELSEWAIT120
473 I=VAL(I$):PLOT10,26,P1$(I):IFI<8THEN474ELSEPLOT13,26,"
 ":GOTO471
474 IFP%(I)<1THENPLOT1,26,"You cannot drop what you don´t have":GOT
O470
475 X=INT(RND(1)*15+B):IFX>13THENO=O-1:P%(I)=P%(I)-1ELSE477
476 PLOT1,26,"A Zombie takes the object and eats it!":WAIT150:GOTO2
0
477 FORT=5TO14:IFA1(X%-2,Y%-1,1)=0ANDK%=TTHEN478ELSENEXT:GOTO480
478 PLOT2,26,"A naughty elf takes the object and
479 WAIT200:PLOT6,26,"has put it in a locked box":GOTO483
480 IFI=5ANDK%=25THEN490
481 IFI=6ANDK%=24THEN500
482 A1(X%-2,Y%-1,1)=1
483 W=W-P1%(I):O=O-1:P%(I)=P%(I)-1
484 B%(X%-2,Y%-1,I)=B%(X%-2,Y%-1,I)+1:GOTO20
490 PLOT1,23,"   The elves gladly take the gift":S=S+3
492 PLOT1,24," They give you 2 strength pills which
494 PLOT1,25,"  increases your strength by three.
496 W=W-.1:O=O-1:P%(I)=P%(I)-1:WAIT300:GOTO20
500 PLOT1,23," Well done,the Wizard can now use his
502 PLOT1,24,"  Apple.He gives you 200 gold coins
504 PLOT1,25,"    and a weight reduction pill.":G=G+200:GOTO496
507 REM--------------------------------------------------------------
----------
508 GOSUB1000:IFK%=17ORK%=16THENPLOT7,26,"You are told to clear off
!":GOTO144
509 PLOT14,22,"GET OBJECT":PLOT6,26,"Do you want to search Y/N":PIN
G
510 GETI$:IFI$="Y"THENPLOT3,23,"These are the objects in the area."
ELSE515
511 FORT=5TO14:IFK%=TTHENPLOT3,26,"Do you think I have X-ray eyes?"
:GOTO144
512 NEXT
513 FORT=0TO9:PLOT1,24,STR$(T):PLOT3,24,")":PLOT6,24,K$(T)
514 PLOT27,24,STR$(B%(X%-2,Y%-1,T)):PING:PLOT12,25,"PRESS ANY KEY."
:GETI$:NEXT
515 IFA1(X%-2,Y%-1,1)=0THENPLOT3,26,"        This box is locked
  ":GOTO144
516 WAIT100:GOSUB1000:PLOT1,26,"SELECT :number of object  (A)leave
it
517 PLOT12,25,"                  ":PING:GETI$:IFI$="A"ORI$="a"THEN144
518 PLOT1,26,"                                ":I=VAL(I$)
519 IFI<10THEN520ELSE515
520 IFB%(X%-2,Y%-1,I)<1THENPLOT7,26,"Sorry,none of those here.":GOT
O515
522 IFW+W1(I)>20THENPLOT7,26,"You can´t carry any more.":GOTO515
524 IFP%(I)=P1%(I)THENPLOT7,26,"Can´t carry more of them.":GOTO515
527 IFI=8THENO=O+1:GOTO540ELSEO=O+1
528 IFI=9THEN536
532 B%(X%-2,Y%-1,I)=B%(X%-2,Y%-1,I)-1:W=W+W1(I)
533 FORT=5TO14:IFK%=TTHENA1(X%-2,Y%-1,0)=27ELSENEXT
534 IFK%=26THENA1(X%-2,Y%-1,0)=27
535 P%(I)=P%(I)+1:GOTO20
536 IFA1(X%-2,Y%-1,0)=26THENG=G+32:A1(X%-2,Y%-1,0)=27:GOTO532
538 X=INT(RND(1)*20+10):G=G+X:A1(X%-2,Y%-1,0)=29:GOTO532
540 IFF>17THENPLOT4,26,"Greedy pig.You´re not hungry.":GOTO20
542 F=F+1+RND(1)*2:W=W+W1(I):A1(X%-2,Y%-1,0)=27:S=S+.5
544 PLOT14,26,1:PLOT15,26,12:PLOT16,26,"CHOMP!":SOUND4,100,0:PLAY0,
1,4,1000
546 WAIT250:PLAY0,0,0,0:GOTO20
569 REM--------------------------------------------------------------
----------
570 IFK%=26ORK%=27THENPLOT6,26,"This box is already open.":GOTO144
571 IFK%=17ORK%=16THENPLOT8,26,"You are told to go home.":GOTO144
572 FORT=5TO14:IFK%=TTHEN576ELSENEXT
574 PLOT8,26,"No box here you twerp!":GOTO144
576 IFP%(3)<1THENPLOT11,26,"You have no keys.":GOTO144
586 P%(3)=P%(3)-1:O=O-1
588 GOSUB1000
589 PLOT6,24,"The box springs open and .":MUSIC1,1,1,0:PLAY1,0,1,10
00
590 PLOTINT((38-LEN(E$(K%-5)))/2),25,E$(K%-5):A1(X%-2,Y%-1,1)=1:WAI
T60
592 A1(X%-2,Y%-1,0)=26:WAIT200:GOTO20
619 REM--------------------------------------------------------------
----------
620 GOSUB1000:MUSIC1,1,1,0:PLAY1,1,1,2000:WAIT50
621 IFK%=26ORK%=27THEN570
622 FORT=5TO14:IFT=K%THEN624ELSENEXT
623 PLOT13,26,"No box here.":GOTO144
624 IFS<5THENPLOT5,26,"You are too weak to open it..":GOTO144
625 IFK%=26ORK%=27THEN570
626 IFK%=17ORK%=16THENPLOT9,26,"You must be joking":GOTO144
627 S=S-1:IFS+F>32-BTHENPLOT4,25,"The box groans,turns blue and"ELS
E630
628 PLOT1,26,"threatens to blow this place sky high":A1(X%-2,Y%-1,0
)=29:WAIT150
629 X=RND(1)*5+1:IFX<1.9THENEXPLODE:GOTO246ELSE20
630 X=RND(1)*2+1:IFX<1.5THEN634
632 GOTO588
634 PLOT1,26,"The box smirks,´try harder´it urges!":WAIT150:GOTO20
699 REM--------------------------------------------------------------
----------
700 GOSUB1000
702 PLOT1,23,"Do you really want to quit........Y/N":PING:GETI$
704 IFI$="Y"ORI$="y"THEN800
706 IFI$="N"ORI$="n"THEN144ELSE702
799 REM--------------------------------------------------------------
----------
800 GOSUB1000
802 PLOT1,23,12:PLOT15,24,"GAME OVER"
804 FORT=1TO10:MUSIC1,2,T,0:MUSIC2,4,T+1,15
806 MUSIC3,4,12,0:PLAY2,2,7,1000:NEXT:PLAY0,0,0,0:WAIT100
807 I1=((S+F)-W/2)*10+G:FORT=0TO9:I2=P2%(T)*P%(T):NEXT:I3=I1+I2+G+S
%
808 IFI3>HTHENPLOT7,24,"You have a high score -":PLOT32,24,STR$(I3)
ELSE816
810 WAIT300:CLS:INPUT"Enter name then press <return>";N$
812 IFLEN(N$)>12THENPING:GOTO810
814 H$=STR$(INT(I3)):H=INT(I3):GOTO820
816 PLOT10,24,"Your score was -":PLOT26,25,STR$(I3):WAIT50
820 PLOT11,26,"Another game  Y/N":PING:GETI$
821 FORT=48000TO49080:POKET,RND(1)*7+16:NEXT:WAIT300
822 IFI$="Y"ORI$="y"THEN1700
824 IFI$="N"ORI$="n"THENEXPLODE:CLS:ENDELSE820
849 REM--------------------------------------------------------------
----------
```

```
850 IFA1(X%-2,Y%-1,0)=28ANDU%=1THEN852ELSE150
852 CLS:PLOT9,1,10:PLOT9,2,10:PLOT10,1,1:PLOT10,2,1
854 PLOT11,1,"CONGRATULATIONS":PLOT11,2,"CONGRATULATIONS
856 PLOT1,5,"The teleport machine whisks you home.
858 PLOT2,7,"The king thanks you for the Wumpus.
860 PLOT1,9,"And gives you 1000 gold coins per man
862 PLOT14,11,"surviving.":X=L%*1000
864 PLOT7,15,STR$(X):PLOT15,15,"GOLD COINS
866 G=G+X:T$="01090906040302010101050708060205020504030209"
868 FORT=1TO44STEP2:T1=VAL(MID$(T$,T,2))
870 MUSIC1,3,T1,0:PLAY1,0,1,1000:WAIT100-T*2:NEXT
880 WAIT50:GOTO800
999 REM-------------------------------------------------------
----------
1000 FORT=22TO26:PLOT1,T,"                                      ":N
EXT
1002 XX=FRE(""):RETURN
1499 REM------------------------------------------------------
----------
1500 FORT=0TO4:READX$,Y$:C$(T)=X$:D$(T)=Y$:NEXT
1502 DATA"Disturbed by you,a group of BATS","flap harmlessly past.
1504 DATA"A few ROCKS litter the ground,no","gold here!
1506 DATA"You stumble across a group of","FRIENDLY WARRIORS.
1508 DATA"You meet a lonely and very","dirty TRAMP.
1510 DATA"Lucky blighter!You find a hoard","of GOLD COINS.
1512 FORT=5TO14:C$(T)="You find a small but heavy,locked"
1514 D$(T)="BOX.":NEXT
1516 FORT=15TO33:READX$,Y$:C$(T)=X$:~$(T)=Y$:NEXT
1518 DATA"A rather rude grizzly bear grabs you","and won't let go!
1520 DATA"You stumble upon a group of","UNFRIENDLY WARRIORS.
1522 DATA"You meet a dirty and nasty","UNFRIENDLY TRAMP.
1524 DATA"A squadron of attacking","SUPERBATS appear.
1526 DATA"You've fallen into a BOTTOMLESS","PIT < one life lost >.
1528 DATA"Three ZOMBIES amble towards","you.Watch out!
1530 DATA"A creepy GIANT-SPIDER scuttles","towards you.
1532 DATA"I smell a Wumpus!","A Wumpus is nearby.
1534 DATA"You've found the WUMPUS.","It's starting to wake up.
1536 DATA"An old WIZARD with an","Apple approaches.
1538 DATA"Some small green ELVES","are playing here.
1540 DATA"A small and heavy box","lies open here.
1542 DATA"A small,heavy,open","and empty box is here.
1544 DATA"You are at the TELEPORT station in","the land of the Wild
 Wumpus.
1546 DATA"Nothing of particular interest is","here.A wolf howls in
the hills.
1548 DATA"Several corpses litter the ground.","A blue-headed Galah
calls you.
1550 DATA"You feel a sudden breeze.","A bottomless pit is nearby.
1552 DATA"A big fat WOMBAT crawls to a bush","As he nears you hear
him fart!
1554 DATA"All around you is mushy swamp.","Better get your wellies
on!
1559 REM------------------------------------------------------
----------
1560 FORT=0TO9:READX$:E$(T)=X$:NEXT
1562 DATA"a shiney sword glistens in the box","a black disk lies in
 the box
1564 DATA"a curved dagger falls to the ground","an unusual key ratt
les.
1566 DATA"a bag of marbles is in it","3 bags of dry roasted peanuts
 appear
1568 DATA"a tin of Superbat Repellent is in it","32 gold coins clat
ter
1570 DATA"an armour plate creaks","a dose of Wumpus Sleep Inducer
1579 REM------------------------------------------------------
----------
1580 FORT=46368TO46375:READB:POKET,B:NEXT
1582 DATA63,63,63,63,63,63,63,63
1584 FORT=46376TO46383:READB:POKET,B:NEXT
1586 DATA63,33,33,33,33,33,33,63
1588 FORT=46424TO46431:READB:POKET,B:NEXT
1590 DATA63,33,33,45,45,33,33,63
1599 REM------------------------------------------------------
----------
1600 FORT=0TO7:READY$,B,C,D:P1$(T)=Y$:P1%(T)=B:K$(T)=Y$:W1(T)=C:P2%
(T)=D:NEXT
1602 DATA"Swords...............",1,1,30
1604 DATA"Daggers...............",2,.4,10
1606 DATA"Armour level.........",4,1,25
1608 DATA"Keys..................",3,.15,10
1610 DATA"Marbles..............",2,.15,2
1612 DATA"Superbat Repellent...",2,.15,20
1614 DATA"Black disks..........",2,.15,5
1616 DATA"Doses of W.S.I.......",2,.2,40
1618 K$(8)="Peanuts...............":K$(9)="Gold coins...........":P1
%(8)=1
1620 P1%(9)=10:W1(8)=.01:W1(9)=.5:P2%(8)=1:P2%(9)=1
1639 REM------------------------------------------------------
----------
1640 FORT=0TO9:READX$:L1$(T)=X$:NEXT
1642 DATABANG,POW,OWWWW!,,ARRGGHHH!,OWCH!,,CRACK,GALAH,CRASHHHH!
1699 REM------------------------------ -----------------------
----------
1700 CLS:PLOT11,3,10:PLOT11,4,10:PLOT13,3,3:INK2:PA~ER0:PLOT13,4,3
1702 PLOT14,3,"WUMPUS HUNT":PLOT14,4,"WUMPUS HUNT
1704 PLOT5,8,"A program for GAMES COMPUTING"
1706 WAIT100:SHOOT
1708 FORT=14TO24
1710 PLOTT,3,5:PLOTT,4,5:WAIT99:SHOOT:NEXT
1712 PLOT13,3,6:PLOT13,4,6:PLOT4,8,5
1714 PLOT14,3,"WUMPUS HUNT":PLOT14,4,"WUMPUS HUNT"
1716 T1$="040810040810040202021004"
1718 FORT=1TO24STEP2:T1=VAL(MID$(T1$,T,2))
1720 MUSIC1,3,T1,0:PLAY1,0,1,1000:WAIT90-T*3:NEXT:WAIT50
1799 REM-----------------------------------------~-------------
----------
1800 PING:PLOT9,12,"SELECT:0 to 9,0=easy-":GETI$
1802 B=VAL(I$):U1=0:U%=0:Y1=0:Y2=0:S%=0
1804 S$=STR$(18-B):W$=STR$(8+B):F$=STR$(18-B):G=0:G$="0":S=18-B:W=7
+B:F=18-B
1806 O=0:O$="0":L%=3:X=0:U%=0:PLOT9,21,"PLEASE WAIT A MINUTE":PLOT7
,21,3
1808 PLOT8,21,12:FORT=0TO19:FORR=0TO19:A1(T,R,0)=29:NEXT:NEXT
1810 FORT=0TO19:FORR=0TO19:A1(T,R,1)=0:NEXT:NEXT
1812 FORT=0TO7:P%(T)=0:P$(T)="0":NEXT
1814 FORT=0TO19:FORQ=0TO19:FORR=0TO9:B%(T,Q,R)=0:NEXT:NEXT:NEXT
1816 XX=FRE(0):CALL#E6CA
1829 REM------------------------------------------------------
----------
1830 CLS:ZZ=23:GOSUB2000
1831 PRINTCHR$(129)"W     W     W     W     W     W     W"
1832 FORT=1TOINT((B+2)/2):ZZ=19:GOSUB2000:NEXT
1833 PRINTCHR$(130)"U     U     U     U     U     U     U"
1834 ZZ=28:GOSUB2000:X%=P1+2:Y%=P2+1:Q%=X%:R%=Y%
1835 PRINTCHR$(131)"M     M     M     M     M     M     M"
1836 FORT=1TOB+5:ZZ=17:GOSUB2000:NEXT
1837 PRINTCHR$(132)"P     P     P     P     P     P     P"
1838 FORT=1TO14-B:ZZ=2:GOSUB20~0:NEXT
1839 PRINTCHR$(133)"U     U     U     U     U     U     U"
1840 FORT=1TOB+5:ZZ=16:GOSUB2000:NEXT
1841 PRINTCHR$(134)"S     S     S     S     S     S     S"
1842 FORT=1TO30:ZZ=0:GOSUB2000:NEXT
1843 PRINT
1844 FORT=1TO30:ZZ=1:GOSUB2000:NEXT
1845 PRINTCHR$(129)"H     H     H     H     H     H     H"
1846 FORT=1TOB+5:ZZ=21:GOSUB2000:NEXT
1847 PRINTCHR$(130)"U     U     U     U     U     U     U"
1848 FORT=1TO30:ZZ=4:X=1:GOSUB2000:NEXT
1849 PRINTCHR$(131)"N     N     N     N     N     N     N"
1850 FORT=1TO18-B:ZZ=3:GOSUB2000:NEXT
1851 PRINTCHR$(132)"T     T     T     T     T     T     T"
1852 FORT=1TOB+9:ZZ=18:GOSUB2000:NEXT
1853 PRINT
1854 FORT=1TO15:ZZ=15:GOSUB2000:NEXT
1855 PRINTCHR$(133)"B     B     B     B     B     B     B"
1856 FORT=1TOB+9:ZZ=20:GOSUB2000:NEXT
1857 PRINTCHR$(134)"Y     Y     Y     Y     Y     Y     Y"
1858 FORT=1TO5:ZZ=5:X=1:GOSUB2000:NEXT
1859 PRINT
1860 FORT=1TO5:ZZ=6:X=1:GOSUB2000:NEXT
1861 PRINTCHR$(129)"D     D     D     D     D     D     D"
1862 FORT=1TO5:ZZ=7:X=1:GOSUB2000:NEXT
1863 PRINTCHR$(130)".     .     .     .     .     .     ."
1864 FORT=1TO5:ZZ=8:X=1:GOSUB2000:NEXT
1865 PRINTCHR$(131)"T     T     T     T     T     T     T"
1866 FORT=1TO5:ZZ=9:X=1:GOSUB2000:NEXT
1867 PRINTCHR$(132)".     .     .     .     .     .     ."
1868 FORT=1TO5:ZZ=10:X=1:GOSUB2000:NEXT
1869 PRINTCHR$(133)"W     W     W     W     W     W     W"
1870 FORT=1TO3:ZZ=11:X=1:GOSUB2000:NEXT
1871 PRINTCHR$(134)"R     R     R     R     R     R     R"
1872 FORT=1TO5:ZZ=12:X=1:GOSUB2000:NEXT
1873 PRINTCHR$(129)"I     I     I     I     I     I     I"
1874 FORT=1TO5:ZZ=13:X=1:GOSUB2000:NEXT
1875 PRINTCHR$(130)"G     G     G     G     G     G     G"
1876 FORT=1TO3:ZZ=14:X=1:GOSUB2000:NEXT
1877 PRINTCHR$(131)"H     H     H     H     H     H     H"
1878 FORT=1TO5:ZZ=24:GOSUB2000:NEXT
1879 PRINTCHR$(132)"T     T     T     T     T     T     T"
1880 FORT=1TO5:ZZ=25:GOSUB2000:NEXT
1881 PRINTCHR$(145)
1882 FORT=1TO5:ZZ=32:GOSUB2000:NEXT
1883 PRINTCHR$(151)
1884 FORT=1TO25:ZZ=33:GOSUB2000:NEXT
1885 PRINTCHR$(150):CALL#E804:WAIT250:GOTO70
1999 REM------------------------------------------------------
----------
2000 P1=INT(RND(1)*19):P2=INT(RND(1)*19)
2002 IFZZ=23THENK=22:GOTO2008
2004 IFZZ=19THENK=31:GOTO2008
2006 IFA1(P1,P2,0)=29THENA1(P1,P2,0)=ZZ:GOTO2016ELSE2000
2007 IFA1(P1,P2,0)=29THEN2008ELSE2008
2008 P1=INT(RND(1)*17+1):P2=INT(RND(1)*17+1)
2009 IFA1(P1,P2,0)=29THEN2010ELSE2008
2010 A1(P1,P2,0)=ZZ:A1(P1-1,P2-1,0)=K:A1(P1-1,P2,0)=K
2012 A1(P1-1,P2+1,0)=K:A1(P1,P2-1,0)=K:A1(P1,P2+1,0)=K
2014 A1(P1+1,P2-1,0)=K:A1(P1+1,P2,0)=K:A1(P1+1,P2+1,0)=K
2016 XX=FRE(""):IFX=1THEN2100ELSERETURN
2099 REM------------------------------------------------------
----------
2100 IFZZ=4ORZZ=12THENB%(P1,P2,9)=1
2102 IFZZ=5THENB%(P1,P2,0)=1
2104 IFZZ=6THENB%(P1,P2,6)=1
2106 IFZZ=7THENB%(P1,P2,1)=1
2108 IFZZ=8THENB%(P1,P2,3)=1
2110 IFZZ=9THENB%(P1,P2,4)=1
2112 IFZZ=10THENB%(P1,P2,8)=1
2114 IFZZ=11THENB%(P1,P2,5)=1
2116 IFZZ=13THENB%(P1,P2,2)=1
2118 IFZZ=14THENB%(P1,P2,7)=1
2120 IFZZ=4THENA1(P1,P2,1)=1
2122 X=0:RETURN
```



D Ellis

ORIC OPTIONS

We introduce you to some of the less well-known facilities of the Oric.

How many ORIC owners realise that TELETEXT compatible pictures can be entered directly to their VDU via the keyboard? In fact how many ORIC owners have mastered the 'Double height' text facility? Due to poor documentation in the ORIC manual the answer is probably few, especially amongst newcomers to computing.

EXCITING EFFECTS OF THE ESCAPE KEY

No mention is made in the ORIC manual about the ESC key. Most owners have probably pressed it a few times and come to the conclusion that it serves no function. How wrong they are. The 'ESCAPE CHARACTER ROUTINE' (ECR) is mentioned briefly in the manual with no explanation as to its purpose. 'Escape Character' also appears in Appendix C — the attribute codes. Neat as the attribute chart may be its layout is confusing. Consider the simplified attributes chart which I have drawn up (Figure 1).

The quickest way to understanding the chart is to experiment with the ESC key. Using the TEXT mode press CTRL and L keys to clear the screen and home the flashing cursor to the top left hand corner of the screen. Press the ESC key once. You have now entered the ECR. The next key you press will not be printed to the screen but will be stored in the screen memory locations as an attribute. As the far left column is used to hold the background colour for the whole of that line let us for example try RED for that line.

Referring to the chart the keyboard character for background RED is Q. Press the Q key and the first line should indeed turn to red. The cursor has now advanced to the next position on the line and the ECR is terminated. However, select a foreground colour with a suitable contrast to RED — say WHITE, press the ESC key again, and by referring to the chart, you will see that the keyboard character for a WHITE foreground is G. Press the G key. The cursor advances again, the ECR is terminated and any characters now typed in will be printed in WHITE. The attributes that have been

Fig. 1. Attribute chart.

POKE/PLOT	COLOUR TEXT FORMAT	KEYBOARD CHARACTER
0	BLACK	@
1	RED	A
2	GREEN	B
3	YELLOW FOREGROUND	C
4	BLUE	D
5	MAGENTA	E
6	CYAN	F
7	WHITE	G
8	Single Height Std. Stationery	H
9	Single Height Alt. Stationery	I
10	Double Height Std. Stationery	J
11	Double Height Alt. Stationery	K
12	Single Height Std. Flashing	L
13	Single Height Alt. Flashing	M
14	Double Height Std. Flashing	N
15	Double Height Alt. Flashing	O
16	BLACK	P
17	RED	Q
18	GREEN	R
19	YELLOW BACKGROUND	S
20	BLUE	T
21	MAGENTA	U
22	CYAN	V
23	WHITE	W

set will continue to the end of the line unless another attribute is set further along that line. Note though that when ESC key is pressed a space is left on the line. It would therefore be impossible to print a word where each character changes colour unless there is a space between the characters. A little trickery with inverse characters, however, could accomplish this function.

The memory location for the top left hand corner of the screen is 48040 (not 48000 as the manual states). On pressing the Q key for a RED background the value 17 will be stored at 48040. Similarly a foreground value of 7 will be stored at 48041 by the Q key. Take a PEEK at 48040 and 48041 and the values 17 and 7 should be returned if the foreground/background colours are WHITE/RED. Numbers between 0 and 23 are therefore encoded as attributes and numbers between 32 and 125 are stored as ASCII characters. Numbers 24 and 31 control screen synchronisation and need not concern us here.

Taking the following example of the contents of the first eleven screen locations; 48040 to 48050:

48040	17	Background RED
48041	07	Foreground WHITE
48042	65	ASCII 'A'
48043	66	ASCII 'B'
48044	03	Foreground YELLOW
48045	67	ASCII 'C'
48046	16	Background BLACK
48047	12	Single Height FLASHING
48048	68	ASCII 'D'
48049	00	Foreground BLACK
48050	69	ASCII'E'

The display should be as shown in Figure 2.

Note that the 12 in 48047 would cause any remaining text to FLASH, and that E will now show on the display as it is in the same colour as the background. It is however still there. Try POKEing in the numbers to see the results, then POKE 48049 with 7 to reveal the E. It should now be clear that to enter different text formats — flashing, double height, graphics etc — then the escape characters H to O should be entered after the ESCV key. Try pressing ESC and L and any remaining text entered on that line will be 'Flash'. ESC and I will give the alternate (graphic) characters in the STATIONARY mode. ESC and M will give SINGLE HEIGHT FLASHING GRAPHIC characters.

DOUBLE HEIGHT CHARACTERS

DOUBLE HEIGHT characters need some explanation. Here the ORIC manual falls very short of perfect, stating that the characters must be entered on EVEN lines (0,2,4 etc). In fact the exact opposite is the case! They must be entered on ODD lines (1,3,5 etc). The result of getting it wrong means that the top of the character is printed on the bottom line and vice versa — this may prove useful for graphic characters, however. Now position the cursor on an ODD line and press CTRL and D keys. Any escape characters and text now entered will appear on two lines. After setting the foreground/background colours press ESC and J. Text now entered will appear as DOUBLE HEIGHT STATIONERY Standard characters. ESC and N will produce FLASHING DOUBLE HEIGHT Standard characters and K and M, the corresponding GRAPHIC characters. Experiment with ESC and various values and you will realize how simple it all is. The screen can thus be built up to give displays as per TELETEXT.

SAVING

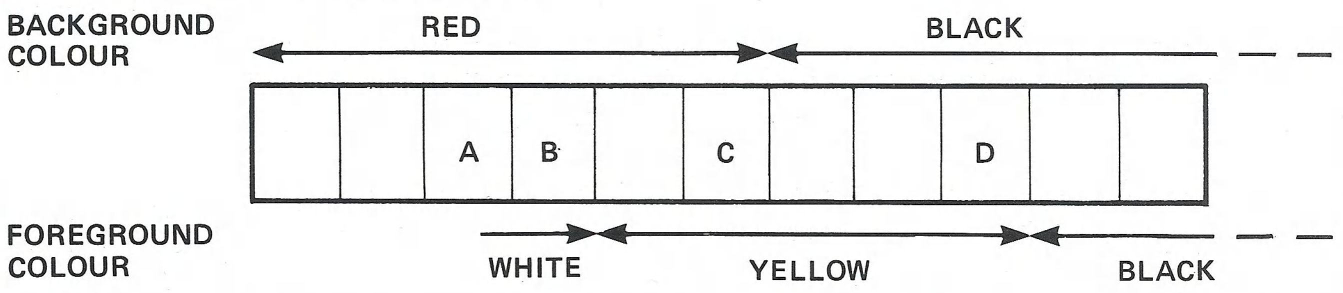
To save the display on tape first turn off the flashing cursor (CTRL & Q) then immobilise the VDU from the keyboard by pressing CTRL and S. This will prevent any keyboard characters that are pressed from appearing on the screen and spoiling the display, although they will still be entered into ORIC as a command to be acted upon. It would also be a good idea to press the RETURN key to clear and SYNTAX errors before typing in the tape saving commmands. However as we will not see what we are typing on the screen, type carefully!

To save the whole of the screen display on tape type:

CSAVE"SCREEN", A48040, E49119 (add ,S for slow mode)

and press RETURN. If the typing has been done carefully, the contents of the screen will be saved onto tape. Now re-enable the VDU (CTRL & S), clear the screen (CTRL & L) then lead the tape by entering: CLOAD"SCREEN" (add, S if saved in SLOW mode) and press RETURN. An exact copy of the original screen should now be displayed. It is not necessary to specify the A and E addresses when CLOADING as this information is already held on the tape. Any part of the screen may be saved by changing the A (start) and E (end) locations.

Fig. 2. Display resulting from example given in text.



SCREEN MEMORY

Why is the start of the screen memory 48040, and not 48000 as the ORIC manual would have you believe? The first line on the screen is only used for displaying 'Loading...', 'Saving...', 'Searching...' and 'Caps'. This is the same line that displays the time, date and page number on the TELETEXT TV. The memory locations for this line are actually 48000 to 48039, but it cannot be assessed through a PRINT or PLOT or PRINT @ statement (using Oric's machine code routine). More of this later.

The only way to use this line is by POKEing directly to it and used thus it is extremely useful. As it is not affected by a CLS or by a scrolling list it can be used very effectively as a STATUS or PROMPT line. It could, for example, be used for list headings. What usually happens once the list reaches the bottom of the screen is that the list scrolls up and the headings disappear from the screen. By POKEing in the headings onto this top line they will remain there permanently until they are changed by another POKE command, irrespective of what happens to the rest of the screen (unless HIRES mode is called).

It is also very useful as a PROMPT line. 'Press SPACE BAR to continue' is a prompt often used. The following subroutine will POKE this message in flashing white characters on a red background to the top line:

```
1000 A$ = "Press SPACE BAR to continue"
1010 POKE 48000,17: POKE 48001,7
1020 POKE 48002,12: P=48004
1030 FOR X= 1 TO LEN(A$)
1040 POKE P+X, ASC(MID$(A$,X,1))
1050 NEXT X
1060 POKE P+X+1,8
1070 RETURN
```

Line 1010 POKEs in background RED (17) and foreground WHITE (7) Line 1020 POKEs in the flashing attribute (12)
Lines 1030 to 1050 POKE the ASCII character of A$ onto the line, and line 1060 POKEs the next location after A$ with the 'Steady' attribute which will stop 'CAPS' from flashing if it is set. If you want to delete the 'CAPS' then POKE 48035 with 1 (RED). This will change the characters to RED, and as the background is RED they will seemingly disappear. They are however still 'there' and can be switched on again by POKEing 48035 with 7 (WHITE). The prompt message can be switched on and off in a similar manner by POKEing 48001 with 1 to make the prompt disappear, and by POKEing with 7 (or any contrasting colour to red) to switch on the prompt message.

If you are familiar with TELETEXT then this is similar to the 'REVEAL' facility where answers to puzzles and questions appear when the 'reveal' button is pressed. The answers are already on the screen but are invisible as the foreground and background colours are the same. Pressing 'Reveal' changes the foreground colour of the answers so making them visible.

CONCLUSION

Hopefully this article will have thrown some light on a subject that is barely dealt with in the ORIC manual. The omission of the 'ESC' key facility is inexcusable, as this is the quickest way of exploring the screen attributes of the ORIC. There will no doubt be a flood of books on the market dealing with programming the ORIC at the BASIC level, as the ORIC manual in general leaves a lot to be desired. Why should an owner have to spend his or her hard earned cash on these expensive books? Apart from specialised subjects like machine code programming, the programming manual supplied with the computer should be far more comprehensive and informative in the first place. The ORIC manual is not the only one to fail in this respect.

If the ORIC computer and its manual had been given to a selection of novices for a few weeks for appraisal before release, then I am sure that from their feedback an error free manual of a much higher standard would have been produced.

TELETEXT

The first part of the article described how to create TELETEXT style displays using ORIC in the immediate mode. How are similar effects produced through a BASIC program? The ORIC manual is not very helpful in this respect. There are four different ways of sending attributes and characters to the VDU.

1 — PRINT
2 — PRINT@(!)
3 — PLOT
4 — POKE

CONTROL MESSAGES

PRINT is, in certain cases, the most cumbersome way of sending attributes and characters to the VDU. For example, to print the message, "DOUBLE FLASH" to the VDU with RED characters and a YELLOW background independent of the foreground/background (F/B) colours that are set globally (all over), the following PRINT statement would be needed:
PRINT CHR$(4);CHR$(27);"S";CHR$27 "N";CHR$(27);"A DOUBLE FLASH";CHR$ (4).
The CHR$ statement will print everything following, onto two lines, including any attributes.CHR$(27) is the 'ESCAPE CHARACTER ROUTINE' (ECR). This will cause the next letter in the print statement to be coded as an attribute; thus "S" in this case would select background yellow, "N" the double height character in flashing mode, and the 'A' at the beginning of the message 'DOUBLE FLASH', would select foreground red. The 'A' itself will not be printed. The CHR$(4) at the end will switch off the double print mode, otherwise all remaining text will be in double print mode.

DOUBLE PRINT AND DOUBLE HEIGHT

Note the difference between

'double print' and 'double height'. 'Double print' prints out the same characters or attributes onto two lines, whereas 'double height' calls a routine which prints out only the top half or bottom half of the character on ONE line (but twice the size). By leaving out the CHR$(4) statement and setting the double height attribute onto two lines, it is possible to print the top half of one character onto the bottom half of another character, which can result in some interesting figures without restoring to redefining characters.

The above message could be held in one string — say P$. Set E$=CHR$ (27) and D$=CHR$(4) and W$="A DOUBLE FLASH", then:

```
P$=D$+E$+"N"+E$+"S'-
'+E$
+W$+D$
```

P$ will then hold all the information required in the one string. If the string needs to be printed to a particular place on the screen then the 'PRINT AT' routine listed in the PRIC manual can be used, and then !X,Y,P$ will print P$ to the X and Y coordinated (see later). It is possible to dispense with the CHR$(2) string if 128 is added to the attribute number. For example the attribute for single height flashing standard characters is 12. Add 128 to this to make 140, then PRINT " "; CHR$ (140); "SINGLE FLASHER" will print 'SINGLE FLASHER' to the screen in flashing single height characters. However, as will be seen shortly, the PLOT command will perform this function with less trouble and with more versatility as to where the message is printed — or should I say plotted.

MEMORY LOCATION 618

Before discussing PLOTting to the screen it would be as well to mention a very useful memory location — 618. This location controls functions on the ORIC depending on the binary BIT pattern that is held there. See Table 1.

With the cursor, VDU and keyclick on, the number stored in 618 will be 1+2+0=3. Note that the keyclick differs from the othrs in that it is turned off when the value of 8 is added to hte number.

This is much easier than the routine given in the ORIC manual, which does not work. The double print mode will need to be switched off again by POKEing 618 with 3 or whatever combination is required. Adding 32 to the POKE number will free the left column, giving a full 40 characters to each line. Location 618 is therefore a very useful way of assessing the various control functions of the ORIC, with the minimum amount of programming.

Table 1. Memory location 618

NUMBER	CONTROL ACTION
1	— CURSOR (on/off)
2	— VDU (enabler/disable)
4	— PRINTER (on/off)
8	— KEYCLICK (off/on)
16	— ESCAPE CHARACTER ROUTINE (on/off)
32	— PROTECTED LEFT COLUMN (on/off)
64	— SDOUBLE PRINT (on/off)

PLOTTING AND 'FLASHING'

When using PLOT the ECR can be dispensed with and the attributes changed from 'S', 'N', and 'A', in the previous example, to their CHR$ numbers, namely 19, 14 and 1. W$ can now be changed to "DOUBLE FLASH" and P$ will become:

```
P$=CHR$(14+CHR$(1)
+CHR$(19)+W$
```

The only problem with PLOTting is the CHR$(4) facility. This will not work, and therefore P$ needs to be PLOTted onto two lines to produce a double height message. Thus:

```
PLOT X,Y,P$ : POT X,Y+1,P$
```

will then produce the same effect as with PRINT or PRINT AT(!), but with less trouble.

It is worth noting that when PLOTting on the ORIC the cursor will not move down to that line which can be a problem if a PRINT statement is to follow. To overcome this, memory location 616 holds the vertical cursor line position, and this can be POKEd with whatever line number the cursor needs to be positioned on (0-26). A 'dummy' PRINT statement will be needed before the actual PRINT statement, which will force the cursor down to the required line — the largest POKE number will therefore be 26, and this would print on the bottom line of screen (26) thus:

```
10 CLS:W$=
"BOTTOM LINE"
20 PLOT 10,10,W$
30 POKE 616,26:PRINT
" "
40 PRINT W$;
50 WAIT 500
```

The 'dummy' PRINT statement is in line 30. If line 30 is left out then W$ will be printed on the top line after the PLOT in line 20. Even more useful are the memory locations 18 and 19. These contain the address that the cursor is positioned at on the screen. By using the DOKE command it is possible to use a PRINT statement that works in the same way as the PLOT statement. Consider the following:

```
10 DEF FN S(X)=
48040+(Y *40)+X
20 CLS:X=10:Y=10
30 DOKE 18,FN S(X)
40 PRINT "DOES IT WORK?"
```

Line 10 DEFines the FuNction S(X), which calculates the screen address given the X and Y co-ordinates. Line 20 clears

▶

to turn off the cursor. Many programmers do not make use of the DEF FN function which is a pity, as it is very useful. I will cover this neglected command in a future article, but for now just accept the fact that it works. By using this method of DOKEing loction 18, the PRINT AT routine in the ORIC manual can be dispensed with.

Returning to the subject of plotting and the example of plotting P$, there is still a problem to ovecome if the string contains its own F/B colours. This problem is that these attributes will continue to the end of the line, and any text on the rest of the line will take on the colour of the string. What is needed is a string which will give a 'boxed' effect without affecting the other colours that are set on the screen. Let us assume that if the F/B colours at the beginning of the line are 0 and 23 respectively, Figure 3 represents the string needed.

Figure 3

8 = Single Height Standard Characters.
1 = RED Foreground.
19 = YELLOW Background.
W$ =The string to be PLOTed.

0 =
23 = F/B colours at beginning of line.

PEEKING FB COLOUR

Now all that is needed is to take a PEEK at the F/B colours set at the beginning of the PLOT line, and add these to the end of the string. The best way is, once again, to DEFine a FuNction for each colour. Thus for the background colour:

```
DEF FN B(X)=PEEK
(48040+(Y *40))
```

and for the foreground colour:

```
DEF FN(X)=PEEK
(48041+(Y *40))
```

To add the functions to the end of P$ would then give:

```
P$=CHR$(8)+CHR$
(1)+CHR$(19)+W$+CHR$(FN
F(Y))+CHR$(FN B(Y))
```

This may appear rather formidable, and for just a few strings would probably not be worthwhile, but in a large program with many strings to be PLOTted it could be reduced into a subroutine giving great flexibility, as in the folowing example.

First set up a subroutine to set up a string array of the attributes and to DEFine the FuNctions:

```
1000 DIM A$(23)
1010 FOR X=1 TO 23
1020 A$(X)=CHR$(X)
1030 NEXT X
1040 DEF FN
B(X)=PEEK(48040+(Y *40))
1050 DEF FN F(X)=
     PEEK(48041+(Y *40))
1060 RETURN
```

The calling subroutine to PLOT the string would then be:

```
2000 P$=A$(M)+
A$(F)+A$(B)+W$+
CHR$(FN F(Y))+
CHR$(FN B(Y))
2010 PLOT X,Y,P$
2020 IF D=1 THEN PLOT
X,Y+1, P$:D=0
2030 RETURN
```

The following variables would then be passed to the subroutine:
W$ — The string to be PLOTted.
X — Horizontal co-odinates. (0-38)
Y — Vertical co-ordinate (0-26)
F — Foreground colour (0-7)
B — Background colour (16-23)
M — Text mode (8-15)
D — 0=Single height.
1=Double height.

Add the following lines:

```
10 CLS: GOSUB 1000
20 X=10 : Y=5 : F=1 : B=19
: M=8
30 W$="TESTING": GOSUB
2000
99 END
```

This would print 'TESTING' in RED letters on a YELLOW background at the X and Y co-ordinates. To test the FuNctions that were added to the end of P$ in line 2000 add:

```
40 PLOT 30,Y,"YES"
```

When the program is run the word 'YES' will be printed after the word 'TESTING', and its colour will be that which is set at the beginning of the line. Add the following lines:

```
50 X=5 : Y=11 : F=2 : B=21
: M=14 : D=1
60 W$="DOUBLE FLASH" :
GOSUB 2000
```

This will print 'DOUBLE FLASH' in flashing Green Double Height figures on a Magenta background, at the X and Y co-ordinates.
Add:

```
70 PLOT 30,Y,"NO"
```

and run the program again.

TROUBLE SHOOTING

Now something is obviously wrong. Although the word 'NO' has been printed after 'DOUBLE FLASH' in the original colour only the top half has been printed, and it is also flashing. Although the colour after the string has been reset, the double height flashing attribute in W$ (M=14) has continued to the end of the line so affecting the word 'NO'. As the MODE attribute before the string is difficult to determine, it is probably better to add the required mode directly to the end of the string — unless of course, no further text is to be printed on that line. In the above example then this would require a change in line 60 to:

```
60 W$="DOUBLE
FLASH"+CHR$(8): GOSUB
2000
```

This would then set any remaining text on that line back to the single height stationary mode.

The subroutine at 2000 should include checks to see that the values passed to the subroutine are within limits and also that the string will not 'overflow' onto the next line. Adding the length of P$ to the X co-ordinate will give the position of the end of the string. As the last X position on the ORIC is 38 then adding:
2005 IF LEN (P$) + X > 38 THEN
(and add the action required) will trap any string from overflowing.

By changing the values in lines 20 and 50, the ease with which information can be sent to the screen and in any form, can be appreciated. Similar results can be achieved by POKEing the ASCII or CHR$ numbers directly to the screen locations from 48000 to 49119. For character strings it is, quite frankly, not worth the effort, as each character has to be stripped from the string by using MID$, and then POKEd to the screen — a far slower method. For individual characters or attributes though, POKEing maybe the better method.

ORIC-1
1 ! 2 @ 3 # 4 $ 5 % 6 ^ 7 & 8 * 9 (Ø) — + = £ \
ESC Q W E R T Y U I O P [{] } DEL
CTRL A S D F G H J K L ; : ' " RETURN
SHIFT Z X C V B N M , < . > ? / SHIFT
← ↓ ↑ →

I stated previously that POKEing was the only way to send characters and attributes to the top line of the scren 48000-48039. There is actually quite a useful way of getting around this problem. The start of the screen location (48000) is held in memory locations 621 and 622. Furthermore, the number of lines on the screen (27) is held in location 623. Very useful effects can be achieved by changing the values in these locations. If 621 is DOKEd with 47960, then the top line of the screen is freed and can be used quite normally. However, a PLOT command will still work on the same lines as previously. It is therefore only possible to use a PRINT statement for this new 'top line'. The bottom line of the screen will not scroll now and could thus be used as the status or prompt line. If the location 623 is POKEd with 28 however, all 28 lines on the screen can now be used quite normally.

CHOOSY SCROLLING

By setting the start address of the screen printing and the number of lines to print, the number of scrolling lines and their position can be chosen at will, thus freezing the remaining parts of the screen for displays or prompts etc. For example, to scroll the middle 10 lines of the screen enter the following program:

```
10 CLS
20 DOKE 621,48320:POKE
623,10
30 PRINT
40 FOR X= 1 TO 500:PRINT
X;:NEXT X
50 GET K$
60 DOKE 621,48000: POKE
623,27
```

Line 20 sets the start of the screen 9 lines down, with 10 scrolling lines. A 'dummy' PRINT statement is used in line 30 which will start the printing at the correct place. Line 40 will show how just the middle 10 lines of the screen are used to print out the X values. Press any key when the printing is finished and this will reset the values to normal. Be careful not to overflow the start and end of screen display as unusual results may occur. If you change the number of lines in line 20 to 27 you will see what I mean!

SCREEN FOOLERY

Once you have set a scrolling area such as this, the 'frozen' parts of the screen can only be printed to by PLOT and POKE. One very exciting prospect is the screen and sets the X and Y co-ordinates. Line 30 DOKEs the screen address into lcoations 18 and 19 using the FuNction S(X), and line 40 will therefore PRINT 'DOES IT WORK' on line 10 of the screen ten spaces from the left. After the program is run the cursor will go back to its original position, in the same way as after a PLOT command. It is a good idea if using this method that it is now possible to set up the HIRES screen and fool the computer into thinking that it is using the TEXT SCREEN. If you enter these two lines:

```
10 HIRES
20 DOKE 621,40920 : POKE
623,199
```

and RUN the program then you will see the flashing cursor at the top left hand corner of the HIRES screen. The value of the ASCII character that you type on the keyboard is what is displayed on the HIRES screen and not the character itself as you will undestand when you read further. For example '?' will produce a continuous line, '@' will produce a blank, ' *' will produce a dotted line. Of course all of the attributes can now be entered directly by using the 'ESC' key and keyboard character, as explained in this article. The number of lines set up is 199 so the cursor can be positioned anywhere on the HIRES screen. Fortunately the standard HIRES cursor is not affected, so all the HIRES commands can be used as normal. HIRES displays could therefore be set up and saved to tape in the same manner as the TEXT screen was. A program could be written to take full advantage of this technique which opens up all sorts of possibilities.

INVERSE CHARACTERS

A similar effect to the plotting routines described earlier can be achieved, albeit with some limitations, by using inverse characters. For some unknown reason Oric omitted a routine to create inverse characters directly from a BASIC keyword, although having the keywords inverse and normal stored in the Oric ROM obviously indicated the possibility of implemented them.

However, it is a fairly simple matter to create an inverse character by adding 128 to its ASCII number, which is encoded by Oric, and printed out as an inverse character. Try the following short program.

```
10 CLS:W$="INVERSE
VIDEO"
20 Y=10:X=5
30 FOR Z=1 TO LEN (W$)
40 PLOT
X+Z,Y,ASC(MID$(W$,Z,1)-
)+128
50 NEXT Z
```

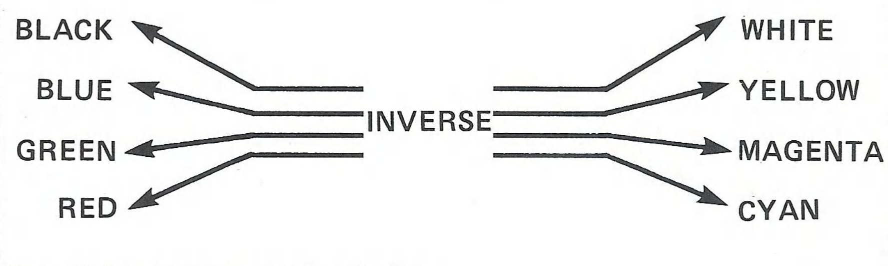


Figure 4. Inverse colours used on the Oric.

Run the program and the string 'INVERSE VIDEO' will be printed in inverse to the screen at the X and Y co-ordinates. The colours will however depend on the F/B attributes that are set to the left of the string; see Figure 4.

If the nearest background attribute to the left of the string is blue, then the string will be printed to the screen in a 'yellow' box. The character colour will be determined by the nearest foreground attribute to the left of the string. If this is white then the characters will be printed in black.

SPEED THROUGH REDEFINITION

A quicker way to achieve inverse characters (if the alternate character set — ASC — is not required) is to redefine them. Unfortunately it is not possible to redefine the whole ASCII character set as this would overflow into the screen memory. However the following program will redefine from ASCII 32 (space), to ASCII 111 (lower case 'o'):

```
10 FOR X=46336 TO 46855
20 POKE X+1024, 63-PEEK(X)
30 NEXT X
```

All that has to be done now when plotting a string is to add the ACS CHR$ number to the beginning of the string. Thus if W$ holds the word to be printed then:

```
PLOT
X,Y,CHR$(9)+W$+CHR$(8)
```

would print W$ as an inverse string — CHR$(9) selecting the ASC and CHR$(8) switching back to the standard set.

It is not possible to print a word where each character changed colour, unless there is a space between the characters, although this could be achieved with some manipulation of the inverse colours. The following program achieves this effect:

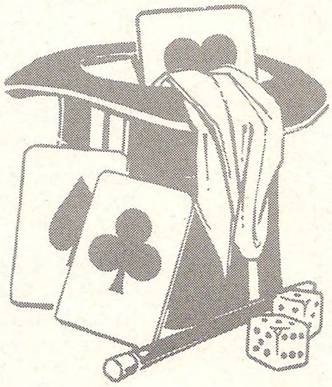
```
10 CLS : W$="ALTERNATE
COLOUR"
20 Y=10 : Y=10
30 FOR Z=1 TO LEN (W$)
STEP 2
40 N=ASC(MID$(WS,Z,1)
50 V=ASC(MID$(W$,Z+1,1)+
128
60 PLOT X+Z,Y,N : PLOT
X+Z+1,Y,V
70 NEXT Z
```

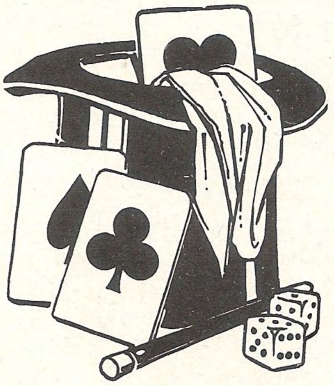
The program plots the characters in W$ alternatively in standard and inverse from and each position therefore changes colour, albeit in the same pattern, showing what can be achieved with a little forethought. Remember to keep the length of W$ to an even number.

Michael Ormerod

ORIC DEMO

If you've just got an Oric, then this program is the ideal place to start. Merely type it in.





```
10 REM ORIC-DEMO
20 REM BY M.E. ORMEROD
30 CLS
40 PRINT
50 PRINT
60 PRINT"  HELLO WELCOME TO ORIC"
70 FOR G=48040 TO 48082 STEP 40
80 POKE G,17
90 NEXT G
100 FOR N=48082 TO 48122 STEP 40
110 POKE N,12
120 NEXT N
130 PRINT
140 PRINT
150 PRINT"  I WILL SHOW YOU WHAT I CAN DO"
160 WAIT 500
170 CLS
180 PRINT"  I CAN MAKE SOUNDS AT A SINGLE COMMAND"
190 PRINT"  YOU CAN TYPE: SHOOT,EXPLODE,ZAP,PING"
200 PRINT"  LET'S TRY THEM, ONE AT A TIME"
210 WAIT 500
220 CLS
230 POKE 48040,18
240 PRINT"EXPLODE"
250 WAIT 100
260 FOR T=1 TO 3
270 EXPLODE
280 WAIT 100
290 NEXT T
300 PRINT"SHOOT"
310 WAIT 100
320 POKE 48080,19
330 FOR N=1 TO 3
340 SHOOT
350 WAIT 100
360 NEXT N
370 PRINT"ZAP"
380 POKE 48120,20
390 FOR R=1 TO 3
400 ZAP
410 WAIT 100
420 NEXT R
430 PRINT"PING"
440 POKE 48160,21
450 FOR K=1 TO 3
460 PING
470 WAIT 100
480 NEXT K
490 WAIT 200
500 CLS
510 PRINT
520 PRINT
530 PRINT"  WATCH THIS FOR A PARTY TRICK"
540 WAIT 500
550 CLS
560 LORES 0
570 FOR D=48040 TO 49119
580 POKE D,INT(RND(1)*7+16)
590 NEXT D
600 PLOT 16,12,"NOT BAD EH!"
610 WAIT 500
620 CLS
630 PRINT
640 PRINT
650 PRINT" HOW ABOUT THIS FOR AN ENCORE!"
660 WAIT 400
670 HIRES
680 FOR N=40960 TO 49709 STEP 40
690 POKE N,INT(RND(1)*7+16)
700 FOR W=0 TO 1100 STEP 40
710 SOUND 1,W/2,0
720 PLAY 1,0,5,5
730 NEXTW,N
740 PLAY 0,0,0,0
750 WAIT 400
760 CURSET 120,100,3
770 CIRCLE 50,1
780 WAIT 200
790 CURSET 90,70,3
800 DRAW 60,0,1
810 DRAW 0,60,1
820 DRAW -60,0,1
830 DRAW 0,-60,1
840 WAIT 300
850 CURSET 0,70,3
860 FILL 60,1,18
870 WAIT 500
880 LORES 0
890 PLOT 16,12,"THANK YOU-GOODBYE."
900 WAIT 300
910 EXPLODE
920 CLS
930 WAIT 1000
940 GOTO 30
```

Shingo Sugiura

SNAKE

Snakes, millipedes, worms – whatever you call them, they're little devils to keep under control!

Your duty as a millipede keeper is to guide the young millipede to the ripe fruits but to make sure he doesn't eat any rocks or part of the wall since he'll get indigestion and you'll be fired from this incredibly competitive job!

When you type in the program, you should save it before you run it. There is a small machine code routine and even though it is very small, a mistake will probably result in a crash which can not be "Reset"ed out of. If you made no mistake the program should first redefine the whole of the upper case font. This was done because I personally hate the spindly characters on the Oric. When the characters have been redefined, the instructions will be printed and will wait until you press the Space bar. When you do press the Space bar, there will be a slight pause while the program initialises various variables then the game begins. You have to be ready on the keys here because as soon as the screen is drawn, the manic millipede will start crunching his way around the screen. You guide him to the fruits whilst avoiding the rocks and the wall. Everytime the young millipede consumes a fruit, he will grow, restricting manoeuvrability even further but if you manage to keep him alive until he becomes an adult millipede, you will go onto look after another millipede who is even more greedy! You have three chances to become an expert millipede keeper however; if you do get fired don't despair, because if your score is good enough, your name will be entered into the hall of fame for posterity.

RUNDOWN

10-40	REM statements.
50	Selects TEXT mode and sets HIMEM to #97FF to protect the characters (this would not be necessary if the Oric's operating system did its job correctly. However, because of a hideous bug, the HIMEM is not set to its correct value on power up).
60	Call a subroutine which assembles the short machine code routine then a subroutine which defines the characters. The machine code routine deals with updating the position of the millipede.
70	Call a subroutine which prints the instructions.
80-330	This is an endless loop which contains the main game.
90	Initialise score, number of lives and number of screens cleared.
100-310	The game itself.
110	Call a subroutine which initialises the position of the millipede and the direction.
120	Call a subroutine which sets up the screen.
130-290	Main loop.
140	Scan keyboard.
150-230	Check which key's been pressed and take appropriate action.
240	Find the ASCII value of the character in front of the millipede's head.
250	It it's not a space, then go and check what it is!
260	Call the machine code routine which moves the millipede.
270-280	Print millipede.
340	Beginning of the subroutine which initialises various variables. This routine is called at the beginning of each game.
350-370	Initialise millipede's position.
380	Define initial direction of millipede.
390	Initialise boolean "DIE" which determines whether millipede is dead or not.
410	End of this subroutine.
420-540	Assemble (or rather, POKE in) the short machine code routine.
550	Beginning of subroutine which prints the instructions at the beginning of the game.
560	Selects TEXT mode and set background colour to black, foreground to white.
570	Get rid of "CAPS" in the top right hand corner of the screen.
580	Get rid of keyclick and flashing cursor.
590-730	Print the instructions.
740	Wait until the Space bar is pressed.
750	End of this subroutine.
760	Beginning of subroutine which sets up the screen.
770	Set back and foreground colours to white and blue respectively (the combination may be changed to suit your own taste).

HOW IT WORKS

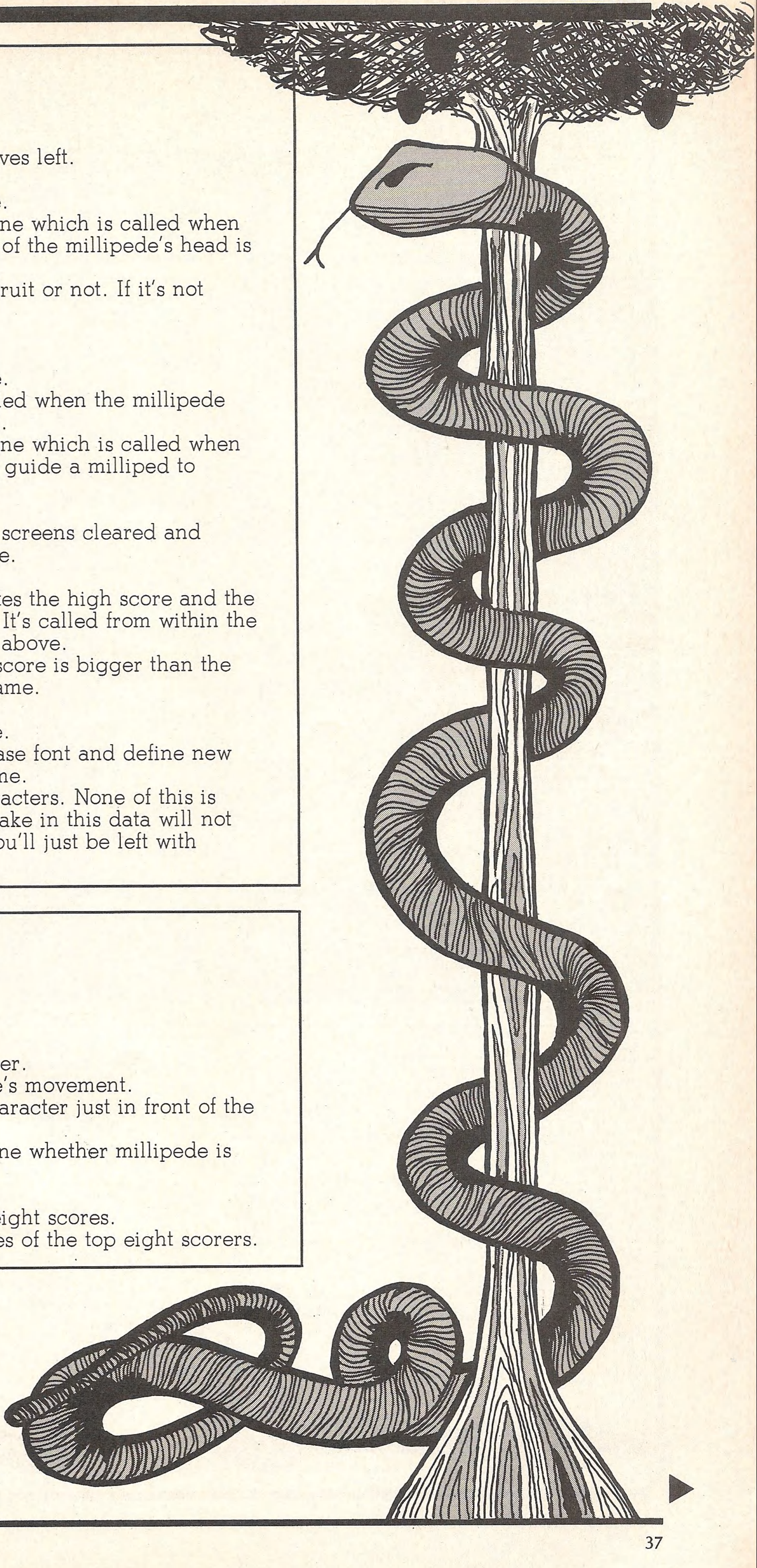
I have tried to use the modular

780-820	Print the wall.
830-860	Print the rocks.
870-900	Print the fruits.
910	Print the score.
920-940	Print the number of lives left.
950-960	Print screen number.
970	End of this subroutine.
980	Beginning of subroutine which is called when the character in front of the millipede's head is not a space.
990	Check whether it's a fruit or not. If it's not DIE = TRUE.
1000	Increment score.
1010	Print new score.
1030	End of this subroutine.
1040-1070	This subroutine is called when the millipede hits a rock or the wall.
1080	Beginning of subroutine which is called when you have managed to guide a milliped to adulthood.
1090-1140	A harmonic Ping!
1150-1170	Increment number of screens cleared and return from subroutine.
1180-1290	Hall of fame.
1310	This subroutine updates the high score and the names of the scorers. It's called from within the subroutine described above.
1320	Check whether your score is bigger than the lowest in the hall of fame.
1330-1540	Input your name.
1550	End of this subroutine.
1560-1590	Redefine the upper case font and define new characters for the game.
1600-1810	Data for the new characters. None of this is machine code so mistake in this data will not crash the machine, you'll just be left with deformed characters.

MAIN VARIABLES

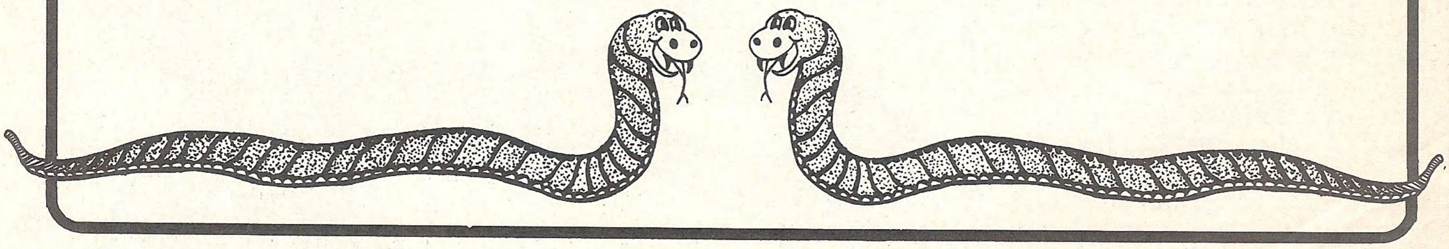
K$	Key pressed.
SC%	Current score.
LI%	Number of lives left.
LE%	Current screen number.
A$	Direction of millipede's movement.
CH%	ASCII code of the character just in front of the millipede's head.
DIE	A boolean to determine whether millipede is dead or not.
LG%	Length of millipede.
FAME	DIM to hold the top eight scores.
NA$	DIM to hold the names of the top eight scorers.

programming techniques used in "Structured languages" in writing "Millipede" so you will find the program relatively easy to understand (in fact, you will find absolutely no dreaded GOTO statements) but for those who'd like to know how it works in detail, a complete run-down is given.



▶

```
10 REM MILLIPEDE
20 REM By Shingo Sugiura
30 REM July 1984
40 REM
50 TEXT:HIMEM #97FF
60 GOSUB430:GOSUB1560
70 GOSUB550
80 REPEAT
90 SC%=0:LI%=2:LE%=1
100 REPEAT
110 GOSUB340
120 GOSUB770
130 REPEAT
140 K$=KEY$
150 IF K$="Z" THEN A$="LEFT"
160 IF K$="X" THEN A$="RIGHT"
170 IF K$="'" THEN A$="UP"
180 IF K$="/" THEN A$="DOWN"
190 PLOT PEEK(47730+LG%),PEEK(47780+LG%)," "
200 IF A$="LEFT" THEN POKE47730,PEEK(47730)-1
210 IF A$="RIGHT" THEN POKE47730,PEEK(47730)+1
220 IF A$="UP" THEN POKE47780,PEEK(47780)-1
230 IF A$="DOWN" THEN POKE47780,PEEK(47780)+1
240 CH%=SCRN(PEEK(47730),PEEK(47780))
250 IF CH%<>32 THENGOSUB980
260 CALL47700
270 PLOT PEEK(47730),PEEK(47780),">"
280 PLOT PEEK(47730+LG%),PEEK(47780+LG%),">"
290 UNTIL DIE OR LG%>30
300 IF DIE THENGOSUB1040ELSEGOSUB1090
310 UNTIL LI%=-1
320 GOSUB1190
330 UNTIL FALSE
340 REM Initialise
350 FOR X=0 TO 40
360 POKE(47730+X),19:POKE(47780+X),10
370 NEXT
380 A$="RIGHT"
390 DIE=FALSE
400 LG%=1
410 RETURN
420 REM Initialise
430 FOR A=0 TO 17
440 READ B:POKE47700+A,B
450 NEXT
460 DIM FAME(9),NA$(9)
470 FOR A=0 TO 9
480 FAME(A)=250:NA$(A)="SHINGOSOFT"
490 NEXT
500 DATA#A0,#28,#B9,#71,#BA
510 DATA#99,#72,#BA,#B9,#A3,#BA
520 DATA#99,#A4,#BA
530 DATA#88,#D0,#F1,#60
540 RETURN
550 REM
560 TEXT:PAPER0:INK7:CLS
570 POKE#BBA3,0
580 POKE#26A,10
590 PRINT:PRINT
600 PRINTSPC(12);CHR$(130)"MILLIPEDE"
610 PRINT:PRINTSPC(8)CHR$(131)"BY SHINGO SUGIURA"
620 PRINT:PRINT"HELP THE YOUNG MILLIPEDE TO BECOME A"
630 PRINT"BIG HAIRY MILLIPEDE BY GUIDING HIM"
640 PRINT"TOWARDS THE BIG JUICY FRUITS THAT"
650 PRINT"ARE LITTERED AROUND THE GARDEN."
660 PRINT"DON'T GUIDE HIM INTO THE WALL OR THE"
670 PRINT"ROCKS THOUGH, HE WILL GET INDIGESTION!"
680 PRINT:PRINTSPC(13)CHR$(132)"CONTROLS":PRINT
690 PRINTSPC(14)"UP   ..   '"
700 PRINTSPC(12)"DOWN   ..   /"
710 PRINTSPC(11)"RIGHT   ..   X"
720 PRINTSPC(12)"LEFT   ..   Z"
730 PRINT:PRINT:PRINTCHR$(134)CHR$(140)"  PRESS THE SPACE BAR TO CO
NTINUE"
740 REPEAT:UNTIL KEY$=" "
750 RETURN
760 REM
770 PAPER7:INK4:CLS
780 FOR V=2 TO 21
790 PLOT 1,V,"?":PLOT 37,V,"?"
800 NEXT
810 PLOT1,1,"?????????????????????????????????????"
820 PLOT1,22,"?????????????????????????????????????"
830 FOR OB=1 TO LE%*3+10
840 X=RND(1)*32+3:Y=RND(1)*17+3
850 PLOTX,Y,"@"
860 NEXT
870 FOR FO=1 TO 30
880 X=RND(1)*32+3:Y=RND(1)*17+3
890 PLOTX,Y,"="
900 NEXT
910 PLOT2,24,"SCORE"+STR$(SC%):PLOT1,24,1
920 FOR A=0 TO LI%
930 PLOT25+A*2,24,"<"
940 NEXT
950 PLOT24,24,5
960 PLOT2,25,"LEVEL"+STR$(LE%):PLOT1,25,6
970 RETURN
980 REM
990 IF CH%<>61 THEN DIE=TRUE:RETURN
1000 SC%=SC%+10:LG%=LG%+2
1010 PLOT7,24,STR$(SC%)
1020 ZAP
1030 RETURN
1040 REM
1050 SHOOT:LI%=LI%-1
1060 WAIT100
1070 RETURN
1080 REM Cleared a screen
1090 PLAY7,0,0,0
1100 PLOT10,10,CHR$(5)+"SCREEN"+STR$(LE%)+CHR$(5)+"CLEARED"
1110 FOR A=12 TO 0 STEP-1
1120 MUSIC1,4,1,A:MUSIC2,4,5,A:MUSIC3,4,8,A
1130 WAIT12
1140 NEXT A
1150 WAIT200
1160 LE%=LE%+1
1170 RETURN
1180 REM Hall of Fame
1190 PAPER0:INK7:CLS
1200 GOSUB1320
1210 PRINT:PRINTSPC(5)CHR$(131)"MILLIPEDE HALL OF FAME"
1220 FOR A=1 TO 8
1230 PLOT2,A*2+3,STR$(A)+".":PLOT8,A*2+3,STR$(FAME(A)):PLOT20,A*2+3
,NA$(A)
1240 PLOT8,A*2+3,1
1250 NEXT A
1260 PLOT1,23,6:PLOT2,23,12
1270 PLOT4,23,"Press the SPACE BAR to replay."
1280 REPEAT:UNTIL KEY$=" "
1290 RETURN
1300 :
1310 REM Update Scores
1320 IF SC%<FAME(8) THEN RETURN
1330 N$=""
1340 FOR A=8 TO 1 STEP-1
1350 IF SC%>FAME(A) THEN FAME(A+1)=FAME(A):F=A:NA$(A+1)=NA$(A)
1360 NEXT A
1370 FAME(F)=SC%
1380 REM Input name
1390 PLOT11,2,"CONGRATULATIONS"
1400 PLOT11,6,"YOUR GREAT SCORE"
1410 PLOT3,8,"QUALIFIES FOR THE HALL OF FAME"
1420 PLOT8,10,"PLEASE ENTER YOUR NAME"
1430 PLOT9,14,22:PLOT28,14,16
1440 PLOT9,15,22:PLOT28,15,16
1450 PLOT9,16,22:PLOT28,16,16
1460 INK1
1470 REPEAT
1480 GET A$
1490 CALL#FB0B
1500 IF ASC(A$)=127 AND LEN(N$)>=1 THEN N$=LEFT$(N$,LEN(N$)-1)
1510 IF ASC(A$)<>127 AND LEN(N$)<15 THEN N$=N$+A$
1520 IF LEN(N$)<15 THEN PLOT11,15,N$+" " ELSE PLOT11,15,N$
1530 UNTIL ASC(A$)=13
1540 NA$(F)=N$:CLS
1550 RETURN
1560 FOR A=46080+(ASC("0")*8) TO 46087+(ASC("Z")*8)
1570 READB:POKEA,B
1580 NEXT
1590 RETURN
1600 DATA #1C,#32,#32,#32,#32,#32,#1C,#00,#0C,#1C,#0C,#0C,#0C,#0C,#
1E,#00
1610 DATA #1C,#26,#06,#06,#1C,#30,#3E,#00,#1C,#26,#06,#0C,#06,#26,#
1C,#00
1620 DATA #0C,#2C,#2C,#2C,#3E,#0C,#0C,#00,#3E,#30,#3C,#06,#06,#26,#
1C,#00
1630 DATA #1C,#32,#30,#3C,#32,#32,#1C,#00,#3E,#06,#06,#06,#0C,#0C,#
0C,#00
1640 DATA #1C,#32,#32,#1C,#32,#32,#1C,#00,#1C,#26,#26,#1E,#06,#26,#
1C,#00
1650 DATA #00,#00,#08,#00,#00,#08,#00,#00,#00,#00,#08,#00,#00,#08,#
08,#10
1660 DATA #12,#33,#33,#3F,#2D,#3F,#1E,#00,#18,#04,#1E,#3F,#3D,#3D,#
1E,#0C
1670 DATA #1E,#1E,#33,#33,#33,#33,#1E,#1E,#3D,#3D,#00,#2F,#2F,#00,#
3D,#3D
1680 DATA #1C,#3E,#3E,#3E,#3D,#3D,#1E,#00,#1C,#32,#32,#3E,#32,#32,#
32,#00
1690 DATA #3C,#32,#32,#3C,#32,#32,#3C,#00,#1C,#32,#30,#30,#30,#32,#
1C,#00
1700 DATA #3C,#32,#32,#32,#32,#32,#3C,#00,#3E,#30,#30,#3C,#30,#30,#
3E,#00
1710 DATA #3E,#30,#30,#3C,#30,#30,#30,#00,#1C,#32,#30,#30,#36,#32,#
1C,#00
1720 DATA #32,#32,#32,#3E,#32,#32,#32,#00,#0C,#0C,#0C,#0C,#0C,#0C,#
0C,#00
1730 DATA #06,#06,#06,#06,#06,#26,#1C,#00,#32,#32,#36,#3C,#3C,#36,#
32,#00
1740 DATA #30,#30,#30,#30,#30,#30,#3E,#00,#3C,#2A,#2A,#22,#22,#22,#
22,#00
1750 DATA #3C,#32,#32,#32,#32,#32,#32,#00,#1C,#32,#32,#32,#32,#32,#
1C,#00
1760 DATA #3C,#32,#32,#3C,#30,#30,#30,#00,#1C,#22,#22,#22,#2A,#24,#
1A,#00
1770 DATA #3C,#32,#32,#3C,#32,#32,#32,#00,#1C,#32,#30,#1C,#06,#26,#
1C,#00
1780 DATA #3E,#0C,#0C,#0C,#0C,#0C,#0C,#00,#32,#32,#32,#32,#32,#32,#
1C,#00
1790 DATA #32,#32,#32,#32,#32,#12,#0C,#00,#22,#22,#22,#2A,#2A,#36,#
22,#00
1800 DATA #22,#22,#14,#08,#14,#22,#22,#00,#26,#26,#26,#1E,#06,#26,#
1C,#00
1810 DATA #3E,#06,#0E,#3E,#38,#30,#3E,#00
```



ALIEN DROPOUT

Your mission is to defend your home planet from the fleet of aliens.

You are the last surviving defender of the planet GORGON. Your home planet is under attack by a fleet of aliens. The few survivors that are left depend on you!

When you run this program, a brief set of instructions will appear and it will wait for you to press the space bar. When you do the screen will clear and you will see the aliens together with your humble self at the bottom of the screen. To move your ship press 'Z' or 'X' and to fire press the space bar. If you manage to blast all the aliens before any of them reach the ground, another set of aliens will appear even closer to your ship.

When you are killed and your score is in the top 8 you will be asked to input your name. This can be anything up to 15 characters long (in fact the computer will not allow you to input anything longer) and when you do so the top 8 scores together with the scores will be displayed. Press the space bar for another game.

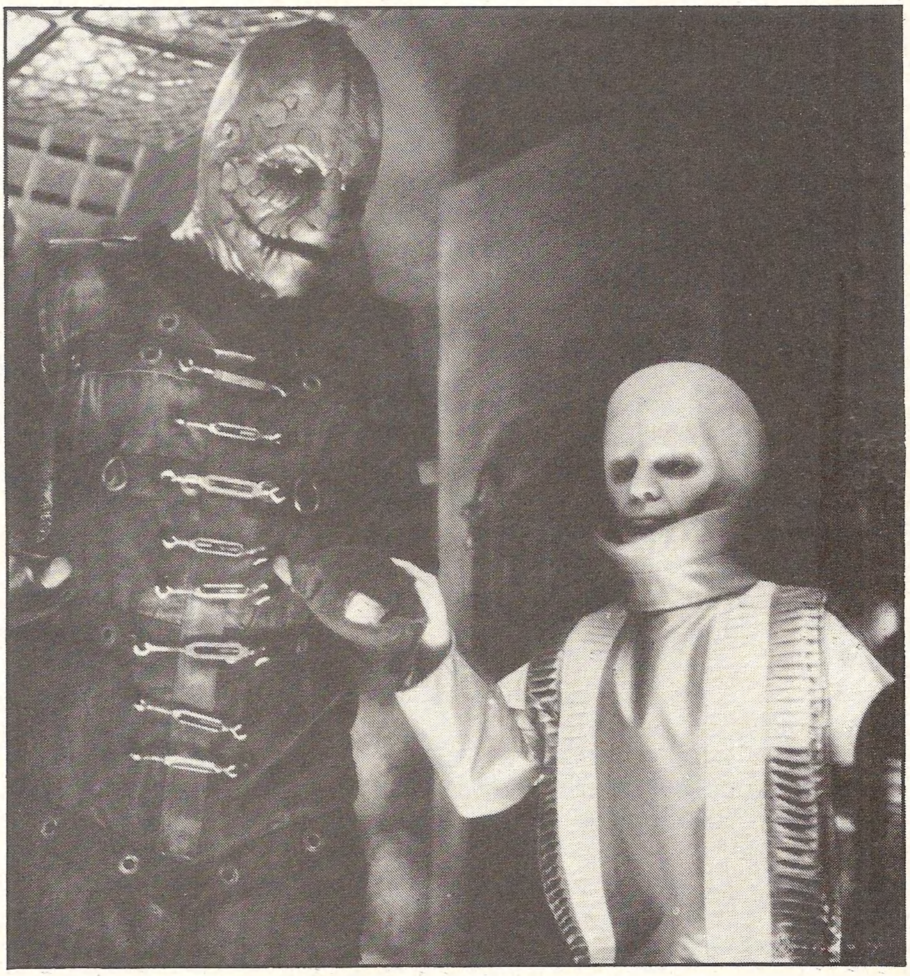
IMPORTANT

When you type in this program replace all £ by a hash character. This is absolutely essential.

CONVERSION CLUES

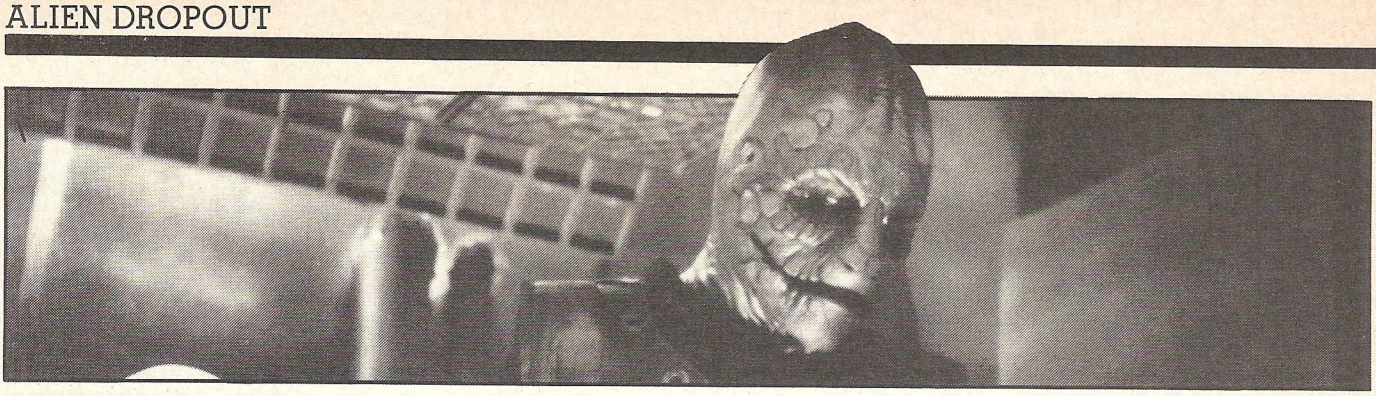
Because this program is entirely written in BASIC and Oric BASIC is fairly standard, it should not be too difficult to convert this program to run on other machines but here are a few hints.

The screen is 40× 26.
PLOT X,Y,A$ is equivalent to PRINT.
TAB(X,Y)A$ or PRINT AT X,Y,A$. Ignore music statement or replace them with appropriate sound commands.
CHR$(4) is double height.

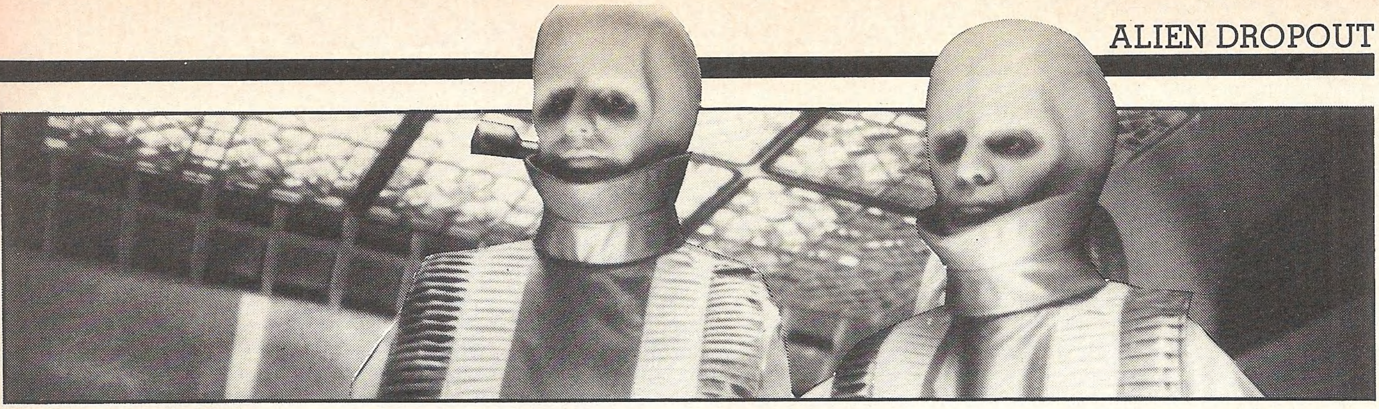


RUNdown

Lines	Action
Lines 1 to 3	REM statements.
Line 10	Selects TEXT mode and resets HIMEM.
Line 20	Defines arrays and print instructions.
Lines 30 to 40	Define variables.
Line 50	Calls the main loop (the game itself).
Lines 150 to 290	Main loop.
Lines 300 to 350	Scan keyboard and take appropriate action.
Lines 400 to 440	Update missile position and print it.
Lines 500 to 540	This procedure is called when an alien is hit.
Lines 810 to 900	Set up the screen.
Lines 1000 to 1160	Print the instructions (called at line 20).
Lines 1250 to 1510	Redefine various characters.
Line 1610	Die.
Lines 1630 to 1710	Hall of fame.
Lines 1940 to 2060	Your score is in top 8.



```
1 REM ALIEN INVASION V2
2 REM Copyright (C) 1983 Oct.
3 REM By Shingo Sugiura
4 :
10 TEXT:GRAB:HIMEM#B400
20 DIM AL(36),SC(9),NA$(9):GOSUB1000
30 HE=4:SC%=0:LE%=0
40 GOSUB700:GOSUB810
50 GOSUB150
60 IF DIE THEN GOSUB 1610:GOTO30
70 GOTO40
144 :
145 REM Main loop
146 :
150 A=4:REPEAT
160 A=A+1:IF A>36 THEN A=4
170 GOSUB300
180 IF B=1 THEN GOSUB 400
190 IF AL(A)=0 OR RND(1)>.85 THEN GOTO 2
80
200 PLOT A,AL(A)," ":AL(A)=AL(A)+1:PLOTA
,AL(A),"!"
210 IF AL(A)>24 THEN DIE=TRUE
220 POKE616,0:POKE617,9:PRINTSC%
280 UNTIL DL%>32 OR DIE
290 RETURN
291 :
299 REM Scan keyboard
300 K$=KEY$
310 X%=X%+(K$="Z" AND X%>3)
320 X%=X%-(K$="X" AND X%<35)
330 IF K$=" " AN~ B=0 THEN MUSIC2,1,7,0:
PLAY2,0,1,500:BX%=X%+1:BY%=
23:B=1
340 PLOTX%,24,SP$
350 RETURN
399 REM Move missile
400 PLOTBX%,BY%," ":BY%=BY%-1
410 IF SCRN(BX%,BY%)<>32 THEN GOSUB 500
420 IF B=1 THEN PLOT BX%,BY%,"¦"
430 IF BY%<3 THEN PLOT BX%,BY%," ":B=0
440 RETURN
499 REM Hit an alien
500 PLOT BX%,BY%,"*":ZAP
510 B=0:AL(BX%)=0
520 DL%=DL%+1:PLOT BX%,BY%," "
530 SC%=SC%+5
540 RETURN
699 REM Initialise
700 X%=20:FLAG=0:DL%=0:SP$=" @ "
710 DIE=FALSE
800 RETURN
809 REM Set up the screen
810 CLS:HE=HE+2:LE%=LE%+1:IF HE%>=12 THE
N HE%=12
811 PLOT0, 5,17:PLOT0,26,18
820 FOR A=0TO1:PLOT0,A,20:PLOT38,A,23:NE
XT:PLOT1,0,1
830 PLOT2,0,"SCORE:":PLOT20,0,"HI-SCORE"
831 H$=RIGHT$(STR$(SC(1)),LEN(STR$(SC(1)
))-1):PLOT29,0,H$
840 L$=RIGHT$(STR$(LE%),LEN(STR$(LE%))-1
):PLOT2,1,"SCREEN"+L$
850 FOR A=4 TO 36:Y=RND(1)*HE+LE%*2:AL(A
)=Y:PLOTA,Y,"!":NEXT A
860 PLOT0,24,5:PLOTX%,24,SP$
870 FOR A=1TO2
880 FOR B=1TO12:MUSIC1,A,B,0:PLAY1,0,2,2
00:WAIT10
890 NEXTB,A
900 B=0:RETURN
999 REM Instructions
1000 CLS:PRINT
1010 PRINTCHR$(4)SPC(10);"ALIEN INVASION
";CHR$(4)
1020 PLOT1,1,10:PLOT1,2,10:PLOT0,1,CHR$(
5):PLOT0,2,CHR$(5)
1030 PRINT:PRINT:PRINTSPC(9);"By Shingo
Sugiura":PLOT0,4,CHR$(1)
1040 PRINT:PRINT" You are the sole defen
der of planet"
1050 PRINT"GORGON. All the others have b
een"
1060 PRINT"destroyed in the previous att
acks"
1070 PRINT"by the aliens. They attack in
 rows so"
1080 PRINT"shooting them is easy, it's a
 matter"
1090 PRINT"of shooting them all."
1100 PRINT" You must defend your home pl
anet as"
1110 PRINT"long as possible as the survi
vors"
1120 PRINT"depend on you."
1130 PRINT:PRINT"CONTROLS:-":PLOT0,16,CH
R$(4)
1140 PRINT:PRINT"'Z'    ---    left."
```

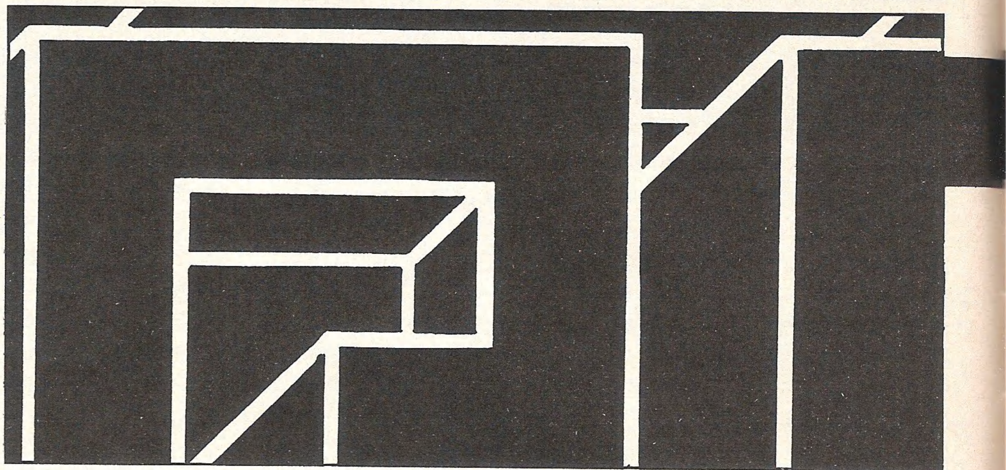


```
1150 PRINT"'X'    ---    right."
1160 PRINT"SPACE ---    fire."
1170 FOR A=46344 TO 46351:READB:POKEA,B:
NEXT A
1180 DATA33,18,30,63,45,63,18,45
1190 FOR A=46592 TO 46599:READB:POKEA,B:
NEXT A
1200 DATA0,12,12,12,45,63,45,63
1210 FOR A=47072 TO 47079:READB:POKEA,B:
NEXT A
1220 DATA0,8,8,8,8,8,8,0
1230 PRINT:PRINTSPC(5);CHR$(140);"Press
the SPACE BAR to play.";:PL
OT0,21,CHR$(2)
1240 REPEAT:UNTIL KEY$=" "
1250 FOR A=46616 TO 46623:READB:POKEA,B:
NEXT A
1260 DATA62,32,32,32,48,48,62,0
1270 FOR A=46632 TO 46639:READB:POKEA,B:
NEXTA
1280 DATA62,32,32,62,48,48,62,0
1290 FOR A=46656 TO 46671:READB:POKEA,B:
NEXT A
1300 DATA34,34,34,62,50,50,50,0
1310 DATA8,8,8,8,24,24,24,0
1320 FOR A=46704 TO 46719:READB:POKEA,B:
NEXT A
1330 DATA62,34,34,34,50,50,50,0
1340 DATA62,34,34,34,50,50,62,0
1350 FOR A=46736 TO 46751:READB:POKEA,B:
NEXT A
1360 DATA60,36,36,62,50,50,50,0
1370 DATA62,32,32,62,6,6,62,0
1380 FOR A=46464 TO 46543:READB:POKEA,B:
NEXT A
1390 DATA62,50,50,50,58,58,62,0
1400 DATA12,4,4,4,12,12,12,0
1410 DATA62,2,2,62,48,48,62,0
1420 DATA62,6,6,62,14,14,62,0
1430 DATA32,32,52,52,62,12,12,0
1440 DATA62,32,32,62,6,6,62,0
1450 DATA62,32,32,62,50,50,62,0
1460 DATA62,2,4,8,24,24,24,0
1470 DATA30,18,18,62,50,50,62,0
1480 DATA62,34,34,62,6,6,62,0
1490 DATA30,18,~8,62,50,50,50,0
1500 FOR A=46600 TO 46607:READB:POKEA,B:
NEXTA
1510 DATA30,18,18,62,50,50,50,0
1520 FOR A=0 TO 8
1530 SC(A)=120-A*10:NA$(A)="Shingosoft"
1540 NEXT A
1550 POKE#26A,10:RETURN
1610 EXPLODE:WAIT200
1620 CLS:GOSUB1900
1630 PRINT:PRINTCHR$(4)SPC(5);"ALIEN INV
ASION HALL OF FAME";CHR$(4)
1640 PLOT0,1,10:PLOT0,2,10:PLOT1,0,17:PL
OT1,1,17:PLOT1,2,17
1650 PLOT3,1,3:PLOT3,2,3:PRINT:PRINT:FOR
 A=1 TO 8
1660 PLOT2,A*2+3,STR$(A)+".":PLOT8,A*2+3
,STR$(SC(A)):PLOT20,A*2+3,N
A$(A)
1670 PLOT8,A*2+3,1:PLOT19,A*2+3,4:NEXT
1680 PLOT5,24,"Press the SPACE BAR to re
play."
1690 PLOT4,24,12:PLOT1,24,6
1700 REPEAT:UNTIL KEY$=" "
1710 RETURN
1900 IF SC%<=SC(8) THEN RETURN
1910 FOR A=8 TO 1 STEP-1
1920 IF SC%>SC(A) THEN SC(A+1)=SC(A):F=A
:NA$(A+1)=NA$(A)
1930 NEXT:SC(F)=SC%
1931 REM Input name.
1940 PRINT:PRINT:PRINT:PRINTSPC(10)CHR$(
4)"!CONGRATULATIONS!"CHR$(4
)
1950 PLOT1,3,10:PLOT1,4,10:PLOT4,3,2:PLO
T4,4,2
1960 PRINT:PRINT:PRINTSPC(10)"Your score
 of ";SC%
1970 PRINT:PRINT"   qualifies for the Ha
ll of Fame."
1980 PRINT:PRINTSPC(8)"Please type your
name."
1990 PLOT11,15,"---------------":N$=""
2000 REPEAT:GET A$
2010 IF ASC(A$)=127 AND LEN(N$)>1 THEN N
$=LEFT$(N$,LEN(N$)-1)
2020 IF ASC(A$)<>127 AND LEN(N$)<15 THEN
 N$=N$+A$
2030 PLOT11,14,N$+" "
2040 UNTIL ASC(A$)=13
2050 NA$(F)=N$:CLS
2060 RETURN
```

N. Sharp

3-D MAZE

It's amazing just how confusing this sort of game can be!



3-D Maze runs on a 16K or 48K Oric-1 computer. In the game, you have the task of getting to a set room in the Maze in as few moves as possible. This room can be identified because when it is in sight, it is shown with a square (which looks like a floormat) lying on the floor. Throughout the game, you see the maze in 3-D as you would if you were actually in the maze. I might also add here that the display given is very realistic (even if I say so myself). When typing in the program, try not to put in any unnecessary spaces (unless you have a 48K model) because the game leaves only about 50 bytes free.

The maze itself is quite simple (see the data in lines 63050 to 63140) but quite small mazes can seem very large to the player when they are only seen in 3-D. The data can easily be changed, but be careful not to run out of memory if you are using a 16K model. A space in the maze is represented by a 0, a 'brick' or 'solid wall' is represented by a 1, and the target is represented by a 2. It is in fact possible to have more than one target, and if more than one is in sight, both will be shown.

When you play the game you are put in the maze in a random position and facing in a random direction. The space bar will move you forward one space, and the left and right cursor keys will turn you through 90 degrees to the left or right.

CONVERSION CLUES

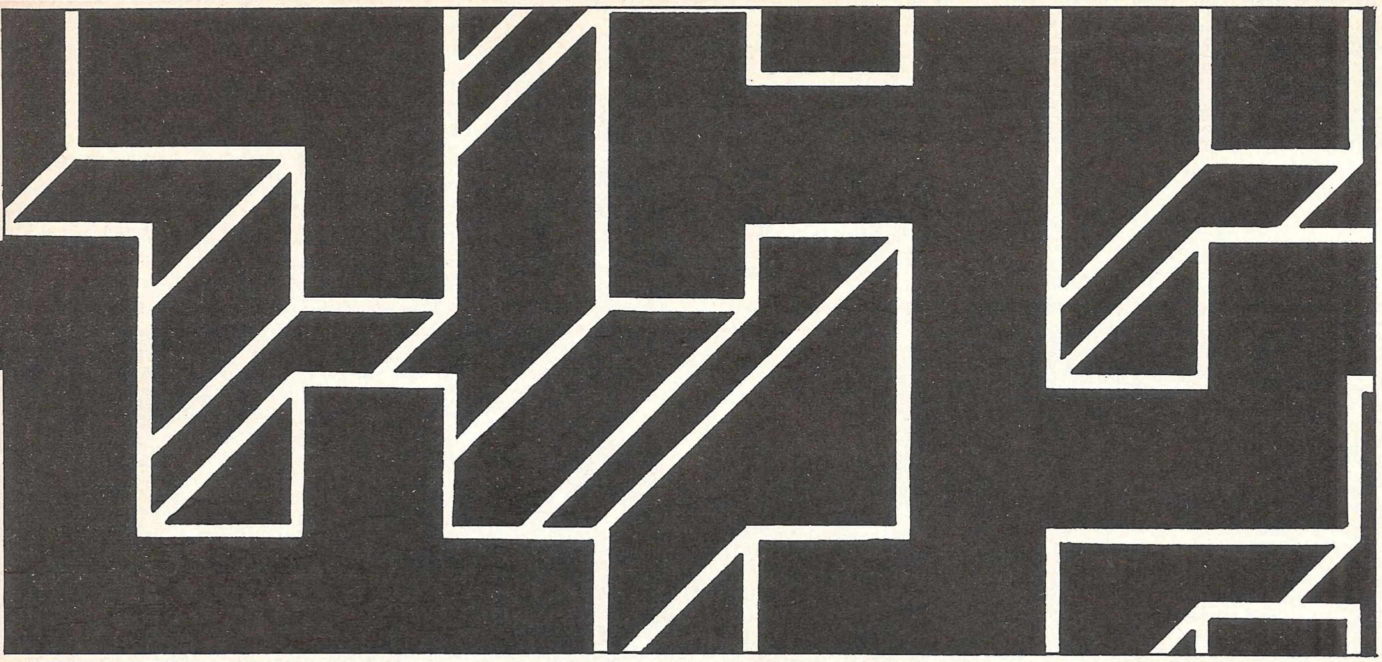
HIRES—puts Oric into High-resolution mode. Clears screen if already High Resolution mode.
TEXT—puts Oric into Text mode. Does not clear screen if already in Text mode.
GET A$—waits for a key to be pressed and thne puts the key pressed into A$.
CURSOR X,Y,FB—puts the Hires cursor at X,Y,FB=1 if you wish to plot a point there, 0 if you wish to rub out a there, 2 if you wish to increase that point or 3 if you wish to leave the point as it is. The Oric's Hires screen measures 240x200 pixels. If your computer has a higher resolution, it will be possible to leave lines 1000-1320 alone, just altering the commands. If not, you will have to change most of the numbers in the routine from 1000 to 1320.
DRAW x,y,fb—Draws a line from current Hires cursor position to a point X pixels right and Y pixels down form the cursor, and moves the cursor to this position.

RUNdown

Lines	Action
1000-1320	Draw routines for separate parts of display.
56000-56060	End routine for when target reached.
51000-57140	Draw display when facing West.
58000-58140	Draw display when facing South.
59000-59140	Draw display when facing East.
60010	GOTO 57000, 58000, 59000 or 60020 depending on direction.
60020-60150	Draw display when facing North.
61000-61150	Check key pressed and perhaps move player.
62000-62999	Randomise position and direction, checking you don't land on a wall.
63000-63150	Set up maze.

VARIABLES USED

MO%	Moves taken so far (turning doesn't count as a move).
A$	General purpose (usually used in GET A$ for inputs such as moving in maze and in end routine).
A%(9,9)	Array holding maze DATA.
X%,Y%	Co-ordinates of player.
D%	Direction of player: 1 North 2 East 3 South 4 West
KE%	ASCII code of key pressed.
A,B	Used in loops for reading maze data.

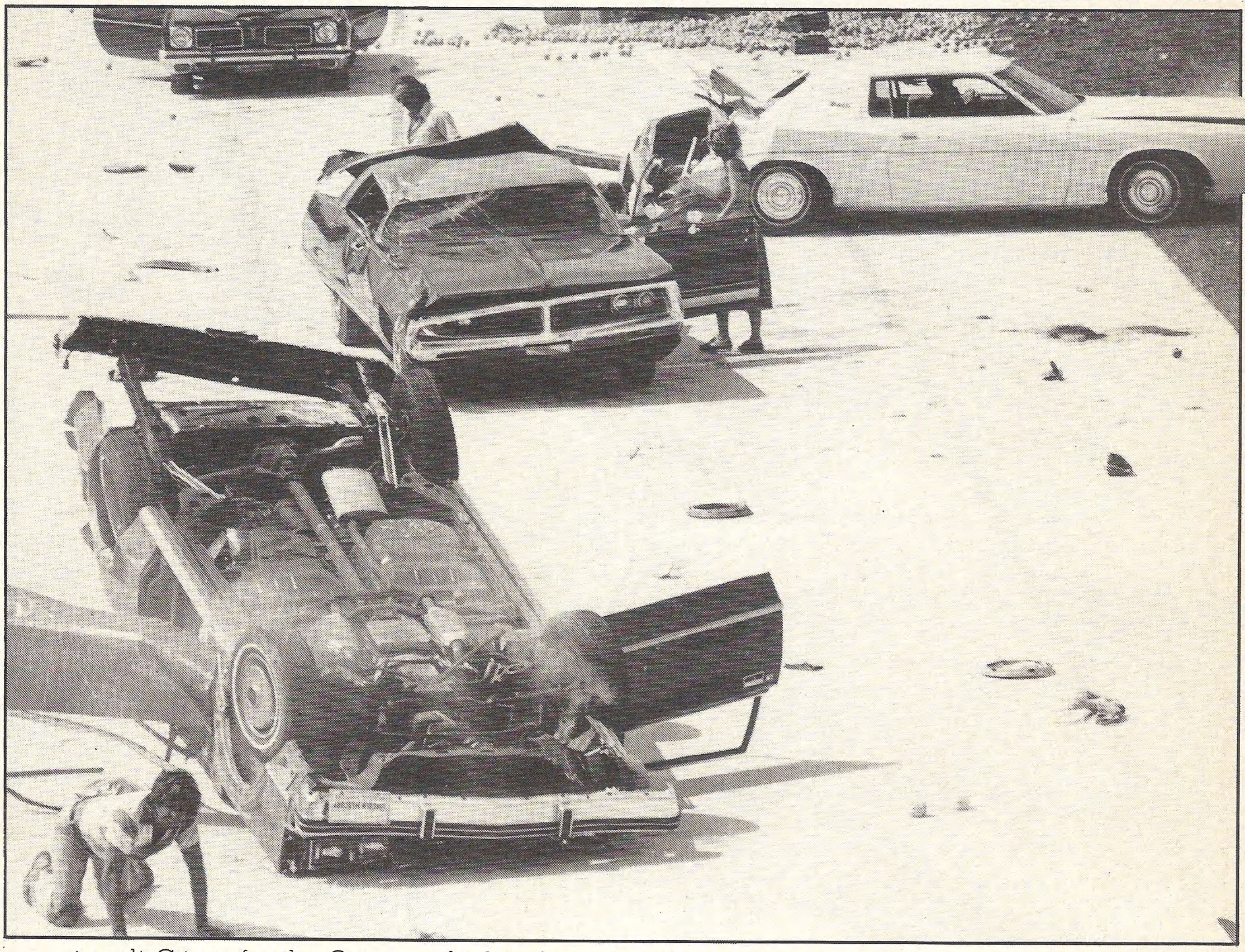


```
120 RELEASE:HIRES:HIMEM #1800
130 GOSUB 63000:GOSUB62000:GOSUB60000
140 GOSUB61000:GOSUB60000:GOTO140
1000 CURSET0,0,1:DRAW20,10,1:CURSET0,120,1:DRAW20,-10,1:RETURN
1010 CURSET0,10,1:DRAW20,0,1:CURSET0,110,1:DRAW20,0,1:RETURN
1020 CURSET20,10,1:DRAW0,100,1:RETURN
1030 CURSET220,10,1:DRAW19,-10,1:CURSET220,110,1:DRAW19,10,1:RETURN
1040 CURSET220,10,1:DRAW19,0,1:CURSET220,110,1:DRAW19,0,1:RETURN
1050 CURSET220,10,1:DRAW0,100,1:RETURN
1060 CURSET20,10,1:DRAW200,0,1:CURSET20,110,1:DRAW200,0,1:RETURN
1070 CURSET20,10,1:DRAW40,20,1:CURSET20,110,1:DRAW40,-20,1:RETURN
1080 CURSET20,30,1:DRAW40,0,1:CURSET20,90,1:DRAW40,0,1:RETURN
1090 CURSET60,30,1:DRAW0,60,1:RETURN
1100 CURSET180,30,1:DRAW40,-20,1:CURSET180,90,1:DRAW40,20,1:RETURN
1110 CURSET180,30,1:DRAW40,0,1:CURSET180,90,1:DRAW40,0,1:RETURN
1120 CURSET180,30,1:DRAW0,60,1:RETURN
1130 CURSET60,30,1:DRAW120,0,1:CURSET60,90,1:DRAW120,0,1:RETURN
1140 CURSET60,30,1:DRAW30,15,1:CURSET60,90,1:DRAW30,-15,1:RETURN
1150 CURSET60,45,1:DRAW30,0,1:CURSET60,75,1:DRAW30,0,1:RETURN
1160 CURSET90,45,1:DRAW0,30,1:RETURN
1170 CURSET150,45,1:DRAW30,-15,1:CURSET150,75,1:DRAW30,15,1:RETURN
1180 CURSET150,45,1:DRAW30,0,1:CURSET150,75,1:DRAW30,0,1:RETURN
1190 CURSET150,45,1:DRAW0,30,1:RETURN
1200 CURSET90,45,1:DRAW60,0,1:CURSET150,75,1:DRAW-60,0,1:RETURN
1210 CURSET90,45,1:DRAW20,10,1:CURSET90,75,1:DRAW20,-10,1:RETURN
1220 CURSET90,55,1:DRAW20,0,1:CURSET90,65,1:DRAW20,0,1:RETURN
1230 CURSET110,55,1:DRAW0,10,1:RETURN
1240 CURSET130,55,1:DRAW20,-10,1:CURSET130,65,1:DRAW20,10,1:RETURN
1250 CURSET130,55,1:DRAW20,0,1:CURSET130,65,1:DRAW20,0,1:RETURN
1260 CURSET130,55,1:DRAW0,10,1:RETURN
1270 CURSET110,55,1:DRAW20,0,1:CURSET110,65,1:DRAW20,0,1:RETURN
1280 CURSET110,55,1:DRAW20,10,1:CURSET130,55,1:DRAW-20,10,1:RETURN
1290 CURSET60,95,1:DRAW120,0,1:DRAW20,10,1:DRAW-160,0,1:DRAW20,-10,
1:RETURN
1300 CURSET90,80,1:DRAW60,0,1:DRAW10,5,1:DRAW-80,0,1:DRAW10,-5,1:RE
TURN
1310 CURSET110,67,1:DRAW20,0,1:DRAW10,5,1:DRAW-40,0,1:DRAW10,-5,1:R
ETURN
1320 CURSET120,62,1:DRAW2,2,1:DRAW-4,0,1:DRAW2,-2,1:RETURN
56000 PING
56010 TEXT:PRINT:PRINT:PRINT" Well done! You reached the target in"
56020 PRINTMO%"moves!"
56030 PRINT"Would you like to play the game again?"
56040 GETA$:IFA$="Y"THEN RUN
56050 IFA$="N"THENEND
56060 GOTO 56040
57000 HIRES
57010 IFA%(X%,Y%)=2THEN56000
57020 IFA%(X%,Y%+1)=1THENGOSUB1000ELSEGOSUB1010
57030 GOSUB1020:IFA%(X%,Y%-1)=1THENGOSUB1030ELSEGOSUB1040
57040 GOSUB1050:IFA%(X%-1,Y%)=1THENGOSUB1060:RETURN
57045 IFA%(X%-1,Y%)=2THENGOSUB1290
57050 IFA%(X%-1,Y%+1)=1THENGOSUB1070ELSEGOSUB1080
57060 GOSUB1090:IFA%(X%-1,Y%-1)=1THENGOSUB1100ELSEGOSUB1110
57070 GOSUB1120:IFA%(X%-2,Y%)=1THENGOSUB1130:RETURN
57075 IFA%(X%-2,Y%)=2THENGOSUB1300:RETURN
57080 IFA%(X%-2,Y%+1)=1THENGOSUB1140ELSEGOSUB1150
57090 GOSUB1190:IFA%(X%-2,Y%-1)=1THENGOSUB1170ELSEGOSUB1180
57100 GOSUB1160:IFA%(X%-3,Y%)=1THENGOSUB1200:RETURN
57105 IFA%(X%-3,Y%)=2THENGOSUB1310
57110 IFA%(X%-3,Y%+1)=1THENGOSUB1210ELSEGOSUB1220
57120 GOSUB1230:IFA%(X%-3,Y%-1)=1THENGOSUB1240ELSEGOSUB1250
57130 GOSUB1260:IFA%(X%-4,Y%)=1THENGOSUB1270:RETURNELSEGOSUB1280
57140 IFA%(X%-4,Y%)=2THENGOSUB1320:RETURNELSERETURN
58000 HIRES
58010 IFA%(X%,Y%)=2THEN56000
58020 IFA%(X%+1,Y%)=1THENGOSUB1000ELSEGOSUB1010
58030 GOSUB1020:IFA%(X%-1,Y%)=1THENGOSUB1030ELSEGOSUB1040
58040 GOSUB1050:IFA%(X%,Y%+1)=1THENGOSUB1060:RETURN
58045 IFA%(X%,Y%+1)=2THENGOSUB1290
58050 IFA%(X%+1,Y%+1)=1THENGOSUB1070ELSEGOSUB1080
58060 GOSUB1090:IFA%(X%-1,Y%+1)=1THENGOSUB1100ELSEGOSUB1110
58070 GOSUB1120:IFA%(X%,Y%+2)=1THENGOSUB1130:RETURN
58075 IFA%(X%,Y%+2)=2THENGOSUB1300
58080 IFA%(X%+1,Y%+2)=1THENGOSUB1140ELSEGOSUB1150
58090 GOSUB1160:IFA%(X%-1,Y%+2)=1THENGOSUB1170ELSEGOSUB1180
58100 GOSUB1190:IFA%(X%,Y%+3)=1THENGOSUB1200:RETURN
58105 IFA%(X%,Y%+3)=2THENGOSUB1310
58110 IFA%(X%+1,Y%+3)=1THENGOSUB1210ELSEGOSUB1220
58120 GOSUB1230:IFA%(X%-1,Y%+3)THENGOSUB1240ELSEGOSUB1250
58130 GOSUB1260:IFA%(X%,Y%+4)=1THENGOSUB1270:RETURN:ELSEGOSUB1280
58140 IFA%(X%,Y%+4)=2THENGOSUB1320:RETURNELSERETURN
59000 HIRES
59010 IFA%(X%,Y%)=2THEN56000
59020 IFA%(X%,Y%-1)=1THENGOSUB1000ELSEGOSUB1010
59030 GOSUB1020:IFA%(X%,Y%+1)=1THENGOSUB1030ELSEGOSUB1040
59040 GOSUB1050:IFA%(X%+1,Y%)=1THENGOSUB1060:RETURN
59045 IFA%(X%+1,Y%)=2THENGOSUB1290
59050 IFA%(X%+1,Y%-1)=1THENGOSUB1070ELSEGOSUB1080
59060 GOSUB1090:IFA%(X%+1,Y%+1)=1THENGOSUB1100ELSEGOSUB1110
59070 GOSUB1120:IFA%(X%+2,Y%)=1THENGOSUB1130:RETURN
59075 IFA%(X%+2,Y%)=2THENGOSUB1300
59080 IFA%(X%+2,Y%-1)=1THENGOSUB1140ELSEGOSUB1150
59090 GOSUB1160:IFA%(X%+2,Y%+1)=1THENGOSUB1170ELSEGOSUB1180
59100 GOSUB1190:IFA%(X%+3,Y%)=1THENGOSUB1200:RETURN
59105 IFA%(X%+3,Y%)=2THENGOSUB1310
59110 IFA%(X%+3,Y%-1)=1THENGOSUB1210ELSEGOSUB1220
59120 GOSUB1230:IFA%(X%+3,Y%+1)=1THENGOSUB1240ELSEGOSUB1250
59130 GOSUB1260:IFA%(X%+4,Y%)=1THENGOSUB1270:RETURNELSEGOSUB1280
59140 IFA%(X%+4,Y%)=2THENGOSUB1320:RETURNELSERETURN
60000 HIRES
60010 OND%GOTO 60020,59000,58000,57000
60020 IFA%(X%,Y%)=2THEN56000
60030 IFA%(X%-1,Y%)=1THENGOSUB1000ELSEGOSUB1010
60040 GOSUB1020:IFA%(X%+1,Y%)=1THENGOSUB1030ELSEGOSUB1040
60050 GOSUB1050:IFA%(X%,Y%-1)=1THENGOSUB1060:RETURN
60055 IF A%(X%,Y%-1)=2THENGOSUB1290
60060 IFA%(X%-1,Y%-1)=1THENGOSUB1070ELSEGOSUB1080
60070 GOSUB1090:IFA%(X%+1,Y%-1)=1THENGOSUB1100ELSEGOSUB1110
60080 GOSUB1120:IFA%(X%,Y%-2)=1THENGOSUB1130:RETURN
60085 IFA%(X%,Y%-2)=2THENGOSUB1300
60090 IFA%(X%-1,Y%-2)=1THENGOSUB1140ELSEGOSUB1150
60100 GOSUB1160:IFA%(X%+1,Y%-2)=1THENGOSUB1170ELSEGOSUB1180
60110 GOSUB1190:IFA%(X%,Y%-3)=1THENGOSUB1200:RETURN
60115 IFA%(X%,Y%-3)=2THENGOSUB1310
60120 IFA%(X%-1,Y%-3)=1THENGOSUB1210 ELSEGOSUB1220
60130 GOSUB1230:IFA%(X%+1,Y%-3)=1THENGOSUB1240ELSEGOSUB1250
60140 GOSUB1260:IFA%(X%,Y%-4)=1THENGOSUB1270:RETURNELSEGOSUB1280
60150 IFA%(X%,Y%-4)=2THENGOSUB1320:RETURN
61000 GET A$:IF ASC(A$)>32 THEN 61000
61010 KE%=ASC(A$)
61020 IFKE%<>8 AND KE%<>9ANDKE%<>32THEN61000
61030 IFKE%=8THEND%=D%-1:IFD%=0THEND%=4
61040 IFKE%=9THEND%=D%+1:IFD%=5THEND%=1
61050 IF KE%=8 OR KE%=9 THEN RETURN
61060 IF KE%<>32 THEN RETURN
61070 IFD%=1ANDA%(X%,Y%-1)=1THEN SHOOT:RETURN
61080 IFD%=2ANDA%(X%+1,Y%)=1THENSHOOT:RETURN
61090 IFD%=3ANDA%(X%,Y%+1)=1THENSHOOT:RETURN
61100 IFD%=4ANDA%(X%-1,Y%)=1THEN SHOOT:RETURN
61105 MO%=MO%+1
61110 IFD%=1THENY%=Y%-1
61120 IFD%=2THENX%=X%+1
61130 IFD%=3THENY%=Y%+1
61140 IFD%=4THENX%=X%-1
61150 RETURN
62000 X%=INT(RND(1)*10):Y%=INT(RND(1)*10):IF A%(X%,Y%)<>0 THEN 6200
0
62010 D%=INT(RND(1)*4)+1
62999 RETURN
63000 DIM A%(9,9):FORA=0TO9:FORB=0TO9:READA%(B,A):NEXTB,A
63050 DATA 1,1,1,1,1,1,1,1,1,1
63060 DATA 1,0,0,1,0,1,0,0,0,1
63070 DATA 1,0,1,0,0,0,1,1,0,1
63080 DATA 1,0,0,0,1,0,0,0,0,1
63090 DATA 1,1,0,1,1,1,1,0,1,1
63100 DATA 1,0,0,1,0,0,1,0,0,1
63110 DATA 1,0,1,1,1,0,1,0,1,1
63120 DATA 1,0,0,0,1,0,1,0,0,1
63130 DATA 1,2,1,0,0,0,0,0,1,1
63140 DATA 1,1,1,1,1,1,1,1,1,1
63150 RETURN
```

A. Gordon

HIRESALT CITY

See how good you are at driving your runner around the city without crashing.



Hiresalt City is for the Oric 1 or Atmos but could be altered to suit any computer because there are no specific PEEKs or POKEs in the program. The game has been written in 'HIRES' and the 'Alt'ernate character set is used to achieve the background.

The game is easy to understand, and although it is very fast, it contains nothing which would confuse someone new to programming.

The object of the game is to drive your runner around the City which appears on the screen without crashing into the perimeter wall, any of the buildings or your own trail which is formed as the runner moves, for 1500 cycles of the program. (One cycle of the program is equivalent to the runner moving one pixel). On completion of 1500 cycles a new City appears with more buildings to avoid.

Each new screen becomes increasingly more difficult until the sixteenth screen when a new type of city appears. This city consists of a perimeter wall but instead of containing buildings it contains vertical lines which are separated from each other by two pixels and by the perimeter walls by one pixel.

VARIABLES USED

A	Position of the runner on the x-axis
B	Position of the runner on the y-axis
C	Varaible used in most of the FOR/NEXT loops
D	The direction of the runner, if: D is 1 then the direction is UP D is 2 then the direction is RIGHT D is 3 then the direction is DOWN D is 4 then the direction is LEFT

I	Variable used in the FOR/NEXT loop of the explosion
L	Number of lives used up
P	Number of points so far raised from completed screens
V	The contents of the next pixel, used to detect a crash
X	The data number which has just been read
Y	The number of buildings to be avoided
Z	The counter within the REPEAT loop

RUNdown

Line	Action
5	Stops the first screen from being one of the harder screens.
10	Defines initial variables and test keys.
20	Decides which direction to alter position in and tests for a crash.
30	Plots the runner and tests for end of the REPEAT loop.
100-170	Alters direction on key being pressed.
200-230	Alters the position of the runner.
300	Explodes the runner and screen changes colour.
305	Finds out if all the lives are used up.
310-320	Ends game.
500	Draws printer.
505	Chooses normal or harder type screen.
520-530	Prints "Hiresalt City" at the top of the screen.
550	Sets up vertical lines of the special screen.

The object of this is to steer yourself around the vertical lines. You will find that speed pressing of the cursor keys is required if you are not to die. Even when the game is not at an end! More cities appear on the screen making each new screen harder than the previous one. The game ends only when you have crashed on three occasions.

The game uses left and right cursor keys only. But it must be borne in mind at all times that left and right apply to the runner and not the user!

While the city is being drawn you will be able to decide which way to go. It is wise to do this because the bottom line is always clear, whereas the path which the runner will automatically take (straight upwards from the centre of the bottom line) has obstacles to avoid.

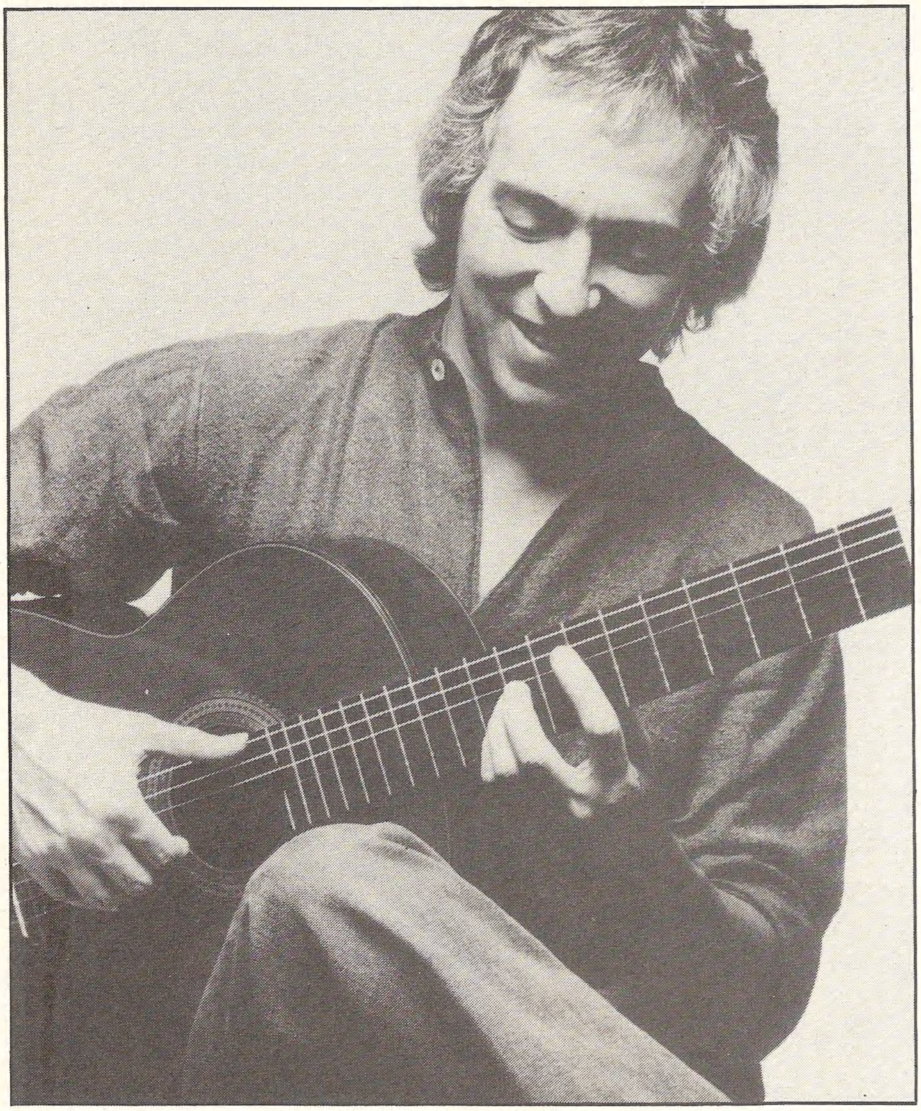
HINT — On the special screens remember that it is possible to turn 180 degrees by pressing the same key twice very quickly.

```
5 IFY=0THENY=Y+1 ELSE IFY>10THENY=Y-1
10 D=1:A=120:B=149:GOSUB500:REPEAT:A$=KEY$:IFA$<>""THENONDGOSUB100,
120,140,160
20 ONDGOSUB200,210,220,230:V=POINT(A,B):IFV=-1 THEN300
30 CURSETA,B,1:Z=Z+1:UNTILZ=1500:P=P+Z:Y=Y+10:PING:Z=0:GOTO5
100 IFA$=CHR$(8)THEND=4 ELSE D=2
110 RETURN
120 IFA$=CHR$(8)THEND=1 ELSE D=3
130 RETURN
140 IFA$=CHR$(8)THEND=2 ELSE D=4
150 RETURN
160 IFA$=CHR$(8)THEND=3 ELSE D=1
170 RETURN
200 B=B-1:RETURN
210 A=A+1:RETURN
220 B=B+1:RETURN
230 A=A-1:RETURN
300 EXPLODE:FORI=1TO7:PAPERI:CIRCLEI,1:WAIT25:NEXTI:WAIT200
305 L=L+1:IFL<3THENGOTO5
310 TEXT:PRINT"YOU ARE DEAD BUT HAVE MANAGED TO SCORE ";P+Z;" POINTS"
320 END
500 HIRES:INK1:CURSET60,50,1:DRAW0,100,1:DRAW120,0,1:DRAW0,-100,1:D
RAW-120,0,1
505 IFY/200=INT(Y/200)THENGOTO550
510 FORC=1TOY+50:CURSETRND(1)*113+60,RND(1)*92+50,1:CHARRND(1)*90+3
2,1,1:NEXTC
520 FORC=1TO104STEP8:READX:CURSET C+70,40,0:CHARX,0,1:NEXTC:RESTORE
:RETURN
530 DATA 72,105,114,101,115,97,108,116,32,67,105,116,121
550 FORC=62TO178STEP3:CURSETC,54,1:DRAW0,92,1:NEXTC:GOTO520
```

David Ellis

CHORD CHORUS

This is for those of you who need to learn about musical chords.



If you are a musician who has been taught to form chords properly read no further. This program will only be of benefit to people who need to learn about chords.

The program will calculate 33 different chord shapes in 12 keys with approximately 7 inversions giving over 2,700 chords. Remember if playing the organ that the root note selected should be played on the pedals, especially with a chord like a ninth which does not contain the root note on the keyboard.

The program contains the most commonly used chord shapes including many of the 'modern' and 'jazz' chords. Some of these may sound rather weird when played on their own but there is a piece of music for every one of them and played in the right phrases they sound just fine. It is possible that you will encounter chords in music books that are not listed in the 'menu'. As many people (even those who should know) are rather ignorant when it comes to the formation of chords, many variants occur, so before you despair, check the following list which gives some of the more common misunderstandings.

C4 This should be read as C SUS. SUS means 'suspend' the third note.

C7-9 This is the same as C7 B9.The '-' stands for flat.

C7 -5 As above. Read as C7 B5.

The + sign is the most commonly misused sign. It means to 'augment' or raise the fifth note by one semi-tone. it is often incorrectly used to mean 'sharpen' the following note. Therefore:

C7+9 This should be read as

C7*9/13 Does not exist, as the 13th, always contains the 9th. Therefore read as 9th.

9th. SUS This is really an 11th chord.

A note following a chord such as in: DM7/G indicates that a G bass pedal is to be played (on the organ) with the chord — although in this case the chord G11 is the same as DM7/G. In fact please note that many chords will appear to have the same notes. The only difference is the root note that would be played on the bass pedals. For example:

C6 and AM7 are the same, as is D11.

C7 B9 is the same as E diminished.

USING THE PROGRAM

All the necessary prompts about what keys to press are displayed on the screen. Chords are selected by pressing A-Z or 1-7 from the menu list of 33 diferent shapes. The root note for the chord is selected next and depending on what this note is, either the natural, sharp or flat of the note is chosen. The program cannot be fooled. F flat for example (which does not exist on the keyboard) will not be accepted. It is not necessary to press the 'RETURN' key when entering characters. The notes will then be displayed on the keyboard and 'played' at the same time. When the notes have been played the 'INVERSION' sign is displayed. Press the 'U' key for the next inversion UP, and the 'D' key for the inversion down. The inversion is protected from running off the screen. Press the space bar to select another chord.

HINTS ON CONVERSION

This program has been written especially for easy conversion to other computers. The sub-routines up to and including 6000 should be typed in, but any line number that ends in a '1' (1001, 3061 etc) should be omitted, as these lines and the sub-routines from 10000 on deal specifically the Oric's graphics and sound. Be careful not to include any spaces in the DATA lines 2030 to 2080. PLOT is Oric's command for printing to the X and Y co-ordinates on the screen, and this will have o be replaced by PRINT AT or whatever commands perform this funtion on your machine. The size of the Oric's screen for plotting is 0-38 in the X axis, 0-26 in the Y-axis. If your machine does not support IF . . . THEN . . . ELSE then you may meed to add a line to get around this. KEY$ is Oric's equivalent to GET$, INKEY$ etc and gets an input from the keyboard. Lines 10-220, which for those of you unfamiliar with structured programming is the program, should be replaced by:

```
 10 GOSUB 1000
 20 GOSUB 2000
 30 GOSUB 3000
 40 GOSUB 4000
 50 GOSUB 5000
 60 PRINT N(1);N(2)-12;
 70 PRINT N(3)-12;
 80 IF CL = 4 THEN PRINT N(4)-12;
 90 PRINT
100 GOSUB 6000
110 GOTO 40
```

Add: 4020 C$= KEY$:IF C$= "" THEN 4020
The numbers that will be printed out are in relation to the keyboard thus:

C	1 or	13	E	5 or	17	Ab	9 or	21
Db	2	14	F	6	18	A	10	22
D	3	15	Gb	7	19	Bb	11	23
Eb	4	16	G	8	20	B	12	24

It will be up to you to design the graphics to represent the keyboard (subs 16000/18000/1900 on Oric) and to play the notes (sub 13000). Sub-routine 14000 may then come in useful as this calculates and plots the notes onto the keyboard. Variable L is set to 5 for sharp or flat notes. Sub-routines 12000 and 12500 calculate the new notes' positions if the inversion of the chord (up or down) is required.

HOW IT RUNS

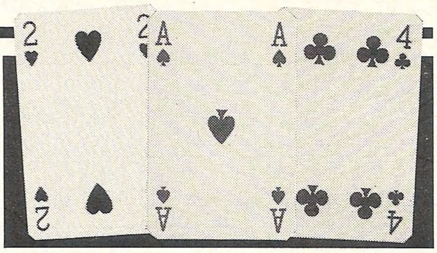
The program itself is contained in lines 10-220. This is just a list of sub-routine calls, apart from line 130 which directs the program depending on the user ressponse to sub-routine 6000. All the sub-routines are fairly short and their purpose clearly stated in the listing.

```
10 GOSUB 1000
20 GOSUB 2000
30 GOSUB 19000
40 GOSUB 18000
50 GOSUB 15000
60 GOSUB 3000
70 GOSUB 16000
80 GOSUB 6000
90 GOSUB 7000
100 GOSUB 9000
110 GOSUB 14000
120 GOSUB 8000
130 ON FLAG GOTO 140,160,180
140 GOSUB 10000
150 GOTO 70
160 GOSUB 12000
170 GOTO 190
180 GOSUB 12500
190 GOSUB 16000
200 GOSUB 9000
210 GOSUB 14000
220 GOTO 120
1000 REM******************************
1005 REM--SET UP VARIABLES/STRINGS---
1010 DIM T$(33),A$(33),AC(33),DN(33)
1011 P1$=" F G A B C D E F G A B C D E F G A B"
1021 SH$=CHR$(12)+CHR$(0)+CHR$(19)
1030 CS$="PRESS A-Z OR 1-7 FOR CHORD SHAPE "
1031 CS$=SH$+CS$+CHR$(16)
1040 CR$="SELECT CHORD ROOT (C,D,E,F,G,A,B) "
1041 CR$=SH$+CR$+CHR$(16)
1051 CL$="                                  "
1060 M0$="'N'(NATURAL) 'S'(#) OR 'B'(FLAT) "
1061 ID$="'D'--DOWN-":IU$="'U'--UP-"
1080 OP$=" PRESS <SPACE BAR> TO CONTINUE "
1081 OP$=SH$+OP$+CHR$(16)
1091 I$="INVERSIONS "
1101 I$=CHR$(0)+CHR$(18)+I$+CHR$(16)
1110 M1$="  'N'(NATURAL) OR 'B'(FLAT)'     "
1111 P2$="122223334"
1120 M2$="  'N'(NATURAL) OR 'S'(SHARP)     "
1130 RETURN
2000 REM******************************
2005 REM----READ IN CHORD DATA-------
2010 FOR X=1 TO 33:READ A$(X):NEXT X
2020 FOR X=1 TO 33:READ T$(X):NEXT X
2025 RETURN
2030 DATAAEH,AEHL,CEHL,ADH,ADHJ,ADHK
2040 DATACDHK,ADG,ADGK,ADHL,ADGJ,AFH
2050 DATAAEI,AEHJ,CEHJ,AEHK,AFHK,AEIK
2060 DATABEIK,AEGK,BEHK,DEHK,CEHK
2070 DATACFHK,CEIK,CEGK,CEHL,CGJK
2080 DATACFHK,CEJK,CFJK,CEIK,CEGK
2090 DATAMAJOR,MAJ 7,MAJ 9,MINOR
2100 DATAMIN 6,MIN 7,MIN 9,MIN B5
2110 DATAMIN7B5,MINMAJ,DIMIN,SUSPEND
2120 DATAAUG (+),6TH.,6/9,7TH.,7 SUS
2130 DATA7 +,7+ B9,7 B5,7 B9,7 #9
2140 DATA9TH.,9 SUS,9TH +,9TH B5
2150 DATA9 MAJ 7,9/13 B5,11TH.
2160 DATA13TH.,13 SUS,13 +,13 B5
3000 REM******************************
3005 REM----PLOT MENU OF CHORDS------
3011 PLOT 1,12,P1$
3020 Z=64
3030 FOR X=1 TO 33
3040 IF X>11 THEN 3080
3050 P$=CHR$(X+Z)+" - "+T$(X)
3060 PLOT 2,X+14,P$
3061 AC(X-1)=0:DN(X-1)=X+14:GOTO 3150
3070 GOTO 3150
3080 IF X>22 THEN 3120
3090 P$=CHR$(X+Z)+" - "+T$(X)
3100 PLOT 15,X+3,P$
3101 AC(X-1)=13:DN(X-1)=X+3:GOTO 3150
3110 GOTO 3150
3120 IF X>26 THEN Q=Z-42 ELSE Q=64
3130 P$=CHR$(X+Q)+" - "+T$(X)
3140 PLOT 28,X-8,P$
3141 AC(X-1)=26:DN(X-1)=X-8
3150 NEXT X
3160 RETURN
6000 REM******************************
6005 REM--GET CHORD/ROOT AND CHECK---
6010 PLOT0,13,CS$
6011 GOSUB 17000
```

```
6030 V=ASC(C$)
6040 IF V<49 OR V>91 THEN 6010
6050 IF V>55 AND V<65 THEN 6010
6060 IF V<65 THEN V=V+42
6070 T=V-65
6071 PLOTAC(T)+1,DN(T),CHR$(17)
6080 PLOT 0,13,CR$
6090 R$=KEY$:IF R$=""THEN 6090
6100 IF R$<"A"OR R$>"G"THEN 6090
6110 IF R$="F"OR R$="C"THEN F$=M2$:GOTO 6140
6120 IF R$="E"OR R$="B"THEN F$=M1$:GOTO 6140
6130 F$=M0$
6140 PLOT 3,13,F$
6150 M$="": M$=KEY$:IFM$=""THEN 6150
6160 IF M$="S" OR M$="B" OR M$="N" THEN 6170 ELSE 6150
6170 IF F$=M1$ AND M$="S"THEN 6150
6180 IF F$=M2$ AND M$="B"THEN 6150
6181 PLOT2,0,CL$:PLOT2,13,CL$
6200 RETURN
7000 REM******************************
7005 REM---CALCULATE NOTE NUMBERS----
7010 RN=ASC(R$)
7020 IF RN<67 THEN RN=(RN-65)*2+9
7030 IF RN>69 THEN RN=(RN-70)*2+5
7040 IF RN>66 AND RN<70 THEN RN=(RN-67)*2
7050 IFM$="S"THEN RN=RN+1
7060 IFM$="B"THEN RN=RN-1
7070 IF V>55 THEN V=V-64 ELSE V=V-22
7080 CL=LEN(A$(V))
7090 FOR X=1 TO CL
7100 N(X)=ASC(MID$(A$(V),X,1))-52+RN
7110 NEXT X
7120 IF M$="S" THEN Y$="#":GOTO 7140
7130 IF M$="B" THEN Y$=CHR$(98) ELSE Y$=""
7140 Q$="CHORD = "+R$ +Y$ +" "+T$(V)
7150 PLOT11,2,Q$
7160 RETURN
8000 REM******************************
8005 REM----PLOT AND GET OPTIONS-----
8010 PLOT 1,13,OP$:FLAG=0
8011 PLOT11,0,I$:PLOT2,1,ID$:PLOT24,1,IU$
8020 G$=KEY$:IF G$=""THEN 8020
8030 IF G$=" "THEN FLAG=1
8041 IF G$="U" THEN FLAG=2
8051 IF G$="D" THEN FLAG=3
8060 IF FLAG=0 THEN 8020
8070 RETURN
9000 REM******************************
9005 REM---CALCULATE PLOT POSITIONS--
9010 FOR X=1 TO CL
9020 IF N(X)<13 THEN P(X)=N(X)+8:GOTO 9070
9030 IF N(X)<18 THEN P(X)=N(X)+9:GOTO 9070
9040 IF N(X)<25 THEN P(X)=N(X)+10:GOTO 9070
9050 IF N(X)<30 THEN P(X)=N(X)+11:GOTO 9070
9060 P(X)=N(X)+12
9070 P(X)=P(X)-12:NEXT X
9080 RETURN
10000 REM*****************************
10005 REM----CLEAR OPTION TITLES-----
10010 PLOT 1,1,CL$:PLOT 1,2,CL$:PLOT2,13,CL$:PLOT2,0,CL
$
10020 PLOT AC(T)+1,DN(T),CHR$(22)
10030 RETURN
12000 REM*****************************
12005 REM---CHORD INVERSION UP-------
12010 SW=N(1)+12
12020 IF SW>36 THEN 12900
12040 N(1)=N(2):N(2)=N(3)
12050 IF CL=3 THEN N(3)=SW:GOTO 12900
12060 N(3)=N(4):N(4)=SW:GOTO 12900
12500 REM*****************************
12505 REM---CHORD INVERSION DOWN-----
12510 IF CL=4 AND N(4)-12<6 THEN12900
12515 IF CL=3 AND N(3)-12<6 THEN12900
12520 IF CL=3 THEN 12600
12530 SW=N(4)-12:N(4)=N(3)
12540 N(3)=N(2):N(2)=N(1):N(1)=SW:GOTO12900
12600 SW=N(3)-12:N(3)=N(2)
12610 N(2)=N(1):N(1)=SW
12900 RETURN
13000 REM*****************************
13005 REM----PLAY KEYBOARD NOTES-----
13020 OC=2:S(X)=N(X)
13030 IF N(X)>12 THEN OC=3:S(X)=N(X)-12
13035 IF N(X)>24 THEN OC=4:S(X)=N(X)-24
13040 PLAY 7,0,0,0
13050 MUSIC 1,OC,S(X),10
13060 WAIT 20
13200 RETURN
14000 REM*****************************
14005 REM---PLOT NOTES ON KEYBOARD---
14010 FOR X=1 TO CL
14020 B=P(X)/2:IF B=INT(B) THEN L=10 ELSE L=5
14030 IF L=5 THEN PLOT P(X),5,"<="
14040 IF L=10 THEN PLOT P(X),10,">?"
14045 GOSUB 13000
14050 NEXT X
14060 WAIT 50:PLAY 0,0,0,0
14070 RETURN
15000 REM*****************************
15005 REM----ORIC SPECIAL POKES------
15010 POKE 48520,20:POKE 48521,7
15020 POKE 48120,22:POKE 48121,0
15030 POKE 48562,12:POKE618,6
15040 POKE 48081,7:POKE 48041,7
15050 POKE 48042,12:POKE 48561,6
15060 POKE 48080,16:POKE 48560,16
15070 POKE 48040,16:POKE 48041,7
15080 FOR X=48600 TO 49080 STEP 40
15090 POKE X,22:POKE X+1,0
15095 POKE X+13,22:POKE X+27,22
15100 NEXT X
15110 FOR X=48000 TO 48039
15120 READ A:POKE X,A
15130 NEXT X
15140 RETURN
16000 REM*****************************
16005 REM----PRINT OUT KEYBOARD------
16020 FOR X=1 TO 9
16030 PLOT 2,X+2,K$(VAL(MID$(P2$,X,1)))
16040 NEXT X
16050 RETURN
17000 REM*****************************
17005 REM---PLAY A LITTLE JINGLE-----
17010 S=30
17020 X=INT(RND(1)*S)
17030 Y=INT(RND(1)*S)
17040 Z=INT(RND(1)*S)
17060 PLAY 7,0,0,0
17070 SOUND 1,X,5
17080 SOUND 2,Y,5
17090 SOUND 3,Z,5
17110 C$=KEY$:IF C$=""THEN 17020
17120 PLAY 0,0,0,0
17130 RETURN
18000 REM*****************************
18003 CLS
18005 REM--KEYBOARD PICTURE STRINGS--
18010 D1$="14343432":D2$=":43432"
18020 D3$=":4343432":D4$=";8989897"
18030 D5$=" 89897":D6$=" 8989897"
18040 D7$=";7":D8$=" 7"
18050 D9$="56":D0$="06"
18060 FOR X=3 TO 11
18070 PLOT 1,X,CHR$(9)
18080 NEXT X
18090 K$(1)=D1$+D2$+D3$+D2$+D3$
18100 K$(2)=D4$+D5$+D6$+D5$+D6$
18110 FOR X=1 TO 17:K$=K$+D8$:NEXT X
18120 K$(3)=D7$+K$
18130 FOR X=1 TO 17:L$=L$+D0$:NEXT X
18140 K$(4)=D9$+L$:PAPER 7:INK 0
18200 RETURN
19000 REM*****************************
19005 REM---REDEFINE CHARACTERS------
19020 X=47488
19030 READ A:IF A=99 THEN 19200
19040 POKE X,A:X=X+1:GOTO 19030
19050 DATA0,0,0,0,0,0,63,63,32,32
19060 DATA32,32,32,32,32,63,1,1,1,1
19070 DATA1,1,1,63,56,56,56,56,56,56
19080 DATA56,63,7,7,7,7,7,7,7,32,32
19090 DATA32,32,32,32,32,63,1,1,1,1,1,1,1
19100 DATA63,1,1,1,1,1,1,1,1,7,7,7,7,7
19110 DATA7,7,7,56,56,56,56,56
19120 DATA56,56,56,63,0,0,0,0,0,0
19130 DATA32,32,32,32,32,32,32,32
19140 DATA7,6,4,0,0,4,6,7,56,24,8,0
19150 DATA0,8,24,56,1,3,7,15,15,7,3,1
19160 DATA33,49,57,61,61,57,49,33,99
19170 DATA21,0,67,32,72,32,79,32,82
19180 DATA32,68,32,32,70,32,73,32,78
19190 DATA32,68,32,69,32,82,7,20,66,89
19200 DATA32,68,65,86,69,32,69,76,76
19210 DATA73,83,32
19220 RETURN
```

TWISTER

Like playing cards? Why not try your hand at playing this version of Pontoon for the Oric?



Twister is a card game which many of you will know better by its traditional names Pontoon or 21. Card sharps amongst you will no doubt like this game as once addicted, always addicted.

It runs on an Oric 1 with 48K of memory. If you are unfamiliar with this game here's a brief rundown of it. the object of Twister is to build up a hand of cards totalling as close to 21 as possible without exceeding it. This version allows the ace to be counted either as one or 11 as you choose. The face cards (Jack, Queen and King) are worth 10 points while all the others are scored by their pips.

In the program the computer takes the bank and deals, which means that to win you must beat the computer's hand. Equalling it just isn't good enough.

The game proceeds as follows:

a) Brief instructions are given.

b) You are asked to input the number of players, (up to six can play) and their names.

c) The computer shuffles the cards and deals one to each player.

d) The players each have a turn to stake some money, you start with £100, the house limit is £499 and the smallest allowed is £1.

e) If the computer particularly likes its first card it can force the players to double their stakes. If a player can't he loses his stake and drops out of that hand.

f) The computer deals a second card to itself and each of the players.

g) Each player then, in turn, has a chance to 'twist' more cards or to 'stick' with what he has, with three exceptions — if your first two cards total 21 then you have a 'natural' and the computer assumes you want to stick, secondly if your last twist took you over 21 you are declared 'bust' and thirdly, if you manage to draw five cards which still total 21 or less you have a '5 & under' which is worth the same as a 21 (except a natural). Anyway you're only allowed five cards!

h) When all players have bust or decided to stick, the computer can then draw cards.

i) The computer collects or pays out according to its and the players' hands, the following grid shows each player's profit or loss; + = profit, − = loss and the numbers refer to multiples of each player's stake as shown in Fig.1.

j) The game repeats from c) until all the players are out.

HOW IT RUNS

The workings of the program are explained within REM statements so I won't duplicate them here but this list of variables will be useful.

HINTS ON CONVERSION

The program has been written as a collection of subroutines, most of which could be transferred either directly or with little modification to most other machines. The three

A$	single character input
ACE	number of aces in a hand
BUST	flag — score greater than 21
CARD%	position of card in hand
CARD$()	the cards
CM$	players' command ie twist or stick
CR$()	players' credits
CS%	value of computer's hand
HND$()	the hands
IN()	flags — players in game
IN	number of players still in game
LEGAL	flag — validity of stake
MS$	message to be displayed
NC%	next card to be dealt
NM$()	players' names
NP%	number of players
PX%(),PY%()	co-ordinates for displaying cards
PY%	amount to collect/pay each player
S$()	values of cards, used in constructing cards
S0%,S1%,T$	temporary variables in shuffle routine
SC%,SC%()	value of hands
ST$,ST%	stake input
ST%()	players' stakes
TS%	top players' score
V%	value of card
EQ$,SP$,TITLE$	display strings
C,I,J,L,P,PP	loop counters

Variables used.

Fig.1

Computer	Player			
	Natural	21 or '5 & under'	under 21	bust
Natural	−1	−2	−3	−3
21 or '5&under'	+2	−1	−2	−2
under 21	+3	+2	−1 or +1	−1
bust	+3	+2	+1	−1

routines which involve graphics — BOARD, DEAL and MESSAGE — however will most certainly have to be rewritten but the layout isn't really crucial. The PEEKs and POKEs used are explained in the REMs, some can be ignored on other machines or alternatives found. The same goes for the MUSIC, PLAY and PING commands. They're not essential but they do add something to the programs.

On those machines without REPEAT UNTIL it may be simulated as follows: replace the REPEAT line with REM REPEAT, replace the UNTIL with IF NOT (condition) THEN GOTO line number of REPEAT eg

```
100 REPEAT        BECOMES        100 REM REPEAT
    •                                •
    •                                •
    •                                •
    •                                •
    •                                •
    •                                •
    •                                •
190 UNTIL                        190 IF A<=10THEN
A >10                            GOTO 100
```

In line 1070 the function SCRN (X,Y) returns the ASCII value of the character in the position with co-ordinates X,Y (NB 0,0 is top left). It could be replaced with a PEEK, the purpose is to check whether or not the 'B' of "Banker" is where it should be, if it isn't then the screen must have scrolled and therefore the board needs to be redrawn.

On the Oric, and most other computers, TRUE=1 ie PRINT 1=1 will print −1, however some computers use 1. In these cases line 4540 must be changed to SC% = SC% + V% 10 *(V%< 2).

```
1 REM ******************
2 REM **    Pontoon    **
3 REM ** © C Hamilton **
4 REM **  6th June 83  **
5 REM ******************
6 REM
7 REM
9 REM ** Goto preliminaries
10 GOSUB10000
99 REM ** Main Loop **
100 REPEAT
110 : GOSUB SETUP     'Initialise
120 : GOSUB SHUFFLE   'Shuffle pack
130 : REPEAT
140 :    GOSUB PREPARE 'Reset variables
149 REM ** Deal first card to each player
150 :    CARD% = 1
159 REM ** Only deal to those players who are still in
160 :    FOR P=0 TO NP%
170 :      IF IN(P) THEN GOSUB DEAL
180 :    NEXT P
190 :    GOSUB STAKE   'Get each player's bet
199 REM ** If banker has a good card then double
200 :    IF VAL(HND$(0,1))<3 THEN GOSUB DOUBLE
209 REM ** Deal second card to each player
210 :    CARD% = 2
220 :    FOR P=0 TO NP%
230 :      IF IN(P) THEN GOSUB DEAL
240 :    NEXT P
250 :    GOSUB TWIST   'Give each player a chance to twist or stick
260 :    GOSUB BANKER  'Banker's turn
270 :    GOSUB PAYOUT  'Pay and collect
280 :    WAIT 300
289 REM ** Count the number of players still in
290 :    IN = 0
300 :    FOR P=1 TO NP%
310 :      IN = IN+IN(P)
320 :    NEXT P
329 REM ** Finish when all out
330 : UNTIL IN=0
339 REM ** Display message
340 : MS$ = "                All out !" :GOSUB MESSAGE
350 : WAIT 200
359 REM ** Offer another game
360 : MS$ = "Another game ? " : GOSUB MESSAGE
370 : REPEAT
380 :    GET A$
390 : UNTIL A$="Y" OR A$="N"
400 UNTIL A$="N"
409 REM ** Restore screen
410 CLS
419 REM ** Clear top line
420 FOR I=#BB80 TO #BBA7
430 : POKE I,32
440 NEXT I
450 END
999 REM **  Get stake from each player **
1000 FOR P=1 TO NP%
1009 REM ** Check if player is still in
1010 : IF NOT IN(P) GOTO 1130
1020 : REPEAT
1030 :    LEGAL = TRUE
1039 REM ** Display credit and request bet
1040 :    MS$ = NM$(P)+" you have £"+MID$(STR$(CR%(P)),2)+" bet = "
1050 :    GOSUB MESSAGE
1059 REM ** Get bet
1060 :    INPUT ST$ : ST% = VAL(ST$)
1069 REM ** Check to see if screen layout has been corrupted
1070 :    IF SCRN(16,1)<>66 THEN GOSUB BOARD
1089 REM ** Check for house limit
1090 :    IF ST%>499 THEN LEGAL = FALSE
1099 REM ** Check that bet is possible
1100 :    IF ST%>CR%(P) OR ST%<1 THEN GOSUB BETBAD
1109 REM ** Repeat until a good bet is given
1110 : UNTIL LEGAL
1119 REM ** Store player's bet and display it beside his name
1120 : ST%(P) = ST% : PLOT 10,PY%(P),ST$
1130 NEXT P
1140 RETURN
1999 REM ** Player twists **
2000 FOR P=1 TO NP%
2009 REM ** Check if player is still in
2010 : IF NOT IN(P) GOTO 2130
2019 REM ** Calculate value of hand so far
2020 : GOSUB COUNT
2029 REM ** Check for 'Natural'
2030 : IF SC%=21 THEN GOSUB NATURAL : GOTO 2130
2039 REM ** Cards so far
2040 : CARD% = 2
2050 : REPEAT
2059 REM ** Offer option twist or stick
2060 :    MS$ = NM$(P)+" twist or stick " : GOSUB MESSAGE
2069 REM ** Get choice and check if layout corrupted
2070 :    INPUT CM$ : IF SCRN(16,1)<>66 THEN GOSUB BOARD
2079 REM ** Check for twist
2080 :    IF CM$="TWIST" OR CM$="T" THEN CARD% = CARD%+1 : GOSUB DEAL
2089 REM ** Recalculate value of hand
2090 :    GOSUB COUNT : SC%(P) = SC%
2094 REM ** Check for 'bust'
2095 :    IF BUST THEN MUSIC 1,0,1,0 : PLAY 1,0,1,800
2100 :    IF BUST THEN MS$ = NM$(P)+" you're bust !" : GOSUB MESSAGE : WAIT200
2109 REM ** Stop when bust or stick or 5 cards in hand
2110 : UNTIL CARD%=5 OR CM$="STICK" OR CM$="S" OR BUST
2119 REM ** Check for 21 and '5 & under' award bonus
2120 : IF (CARD%=5 AND NOT BUST) OR SC%(P)=21 THEN SC%(P) = 100
2130 NEXT P
2140 RETURN
2998 REM ** Banker's play **
2999 REM ** Reveal banker's hand
3000 FOR C=1 TO 2
3010 : PLOT PX%(0,C),PY%(0),MID$(HND$(0,C),2)
3020 NEXT C
3029 REM ** Calculate banker's value
3030 P = 0 : GOSUB COUNT : CS% = SC%
3039 REM ** Check for 'Natural'
3040 IF CS%=21 THEN CS% = 1000 : PING : RETURN
3049 REM ** Decide whether or not to stick
3050 IF CS%>17 AND RND(1)>.1 THEN RETURN
3059 REM ** Find top players' score
3060 TS% = 0
3070 FOR P=1 TO NP%
3080 : IF NOT IN(P) GOTO 3100
3088 REM ** Only consider those hands that aren't bust
3089 REM ** or 'naturals', a 'natural' can only be beaten by a 'natural'
3090 : IF (SC%(P)<22 OR SC%(P)=100) AND SC%(P)>TS% THEN TS% = SC%(P)
3100 NEXT P
3109 REM ** If better than players' top score then stick
3110 IF CS%>=TS% THEN RETURN
3120 P = 0 : CARD% = 2
3130 REPEAT
3139 REM ** Twist a card
3140 : CARD% = CARD%+1 : GOSUB DEAL
3149 REM ** Calculate new value
3150 : GOSUB COUNT : CS% = SC%
3154 REM ** Check for bust
3155 : IF BUST THEN MUSIC 1,0,1,0 : PLAY 1,0,1,800
3160 : IF BUST THEN MS$ = "Banker has bust !" : GOSUB MESSAGE : WAIT 200
3169 REM ** Check for 21 and '5 & under' and award bonus
3170 : IF (CARD%=5 AND NOT BUST) OR CS%=21 THEN CS% = 100
3179 REM ** Stop at 5 cards or bust or better than players' top
3180 UNTIL CARD%=5 OR BUST OR CS%>=TS% OR (CS%>17 AND RND(1)>.1)
3190 RETURN
3999 REM ** Pay out **
4000 FOR P=1 TO NP%
4009 REM ** Check if player is still in
4010 : IF NOT IN(P) GOTO 4200
4019 REM ** Check for draw, banker collects stake
4020 : IF CS%/SC%(P)=1 THEN PY% = -ST%(P) : GOTO 4150
4029 REM ** Check for banker having 'natural'
4030 : IF CS%<1000 GOTO 4070
4039 REM ** If player has 21 or '5 & under' then collect double stake
4040 : IF SC%(P)=100 THEN PY% = -2*ST%(P)
4049 REM ** If player is bust or under 21 then collect treble stake
4050 : IF SC%(P)<100 THEN PY% = -3*ST%(P)
4060 : GOTO 4150
4069 REM ** Check for banker having 21 or '5 & under'
4070 : IF CS%<100 GOTO 4110
4079 REM ** If player has 'natural' then pay treble sta~e
4080 : IF SC%(P)=1000 THEN PY% = 3*ST%(P)
4089 REM If player is bust or under 21 then collect double stake
4090 : IF SC%(P)<100 THEN PY% = -2*ST%(P)
4100 : GOTO 4150
4108 REM ** Banker is bust or under 21
4109 REM ** If player has 'natural' then pay treble stake
4110 : IF SC%(P)=1000 THEN PY% = 3*ST%(P)
4119 REM ** If player has 21 or '5 & under' then pay double stake
4120 : IF SC%(P)=100 THEN PY% = 2*ST%(P)
4129 REM ** If player's hand is closer to 21 then pay stake
4130 : IF SC%(P)>CS% AND SC%(P)<100 THEN PY% = ST%(P)
4139 REM ** If banker's hand is closer to 21 then collect stake
4140 : IF SC%(P)<CS% AND SC%(P)<100 THEN PY% = -ST%(P)
4149 REM ** Display verdict
4150 : IF PY%<0 THEN MS$ = NM$(P)+" owes the bank £"+MID$(STR$(PY%),2)
4160 : IF PY%>0 THEN MS$ = "Banker pays "+NM$(P)+" £"+MID$(STR$(PY%),2)
4170 : GOSUB MESSAGE : WAIT 200
4179 REM ** Pay or collect
4180 : CR%(P) = CR%(P)+PY%
4189 REM ** Check for player out of money
4190 : IF CR%(P)<1 THEN GOSUB BROKE
4200 NEXT P
```

```
4210 RETURN
4498 REM ** Count score on cards **
4499 REM ** Reset counters and flag
4500 SC% = 0 : ACE% = 0 : BUST = FALSE
4509 REM ** Total value of cards
4510 FOR C=1 TO 5
4519 REM ** Check for empty hand
4520 : IF HND$(P,C)="" THEN C = 5 : GOTO 4560
4529 REM ** Find value of card, check for ace
4530 : V% = VAL(HND$(P,C)) : IF V%=1 THEN ACE% = ACE%+1
4539 REM ** Add value of card to total, check for 10 and face cards
4540 : SC% = SC%+V%-(V%<2)*10
4560 NEXT C
4569 REM ** Decide whether aces to be 1 or 11
4570 REPEAT
4579 REM ** If there is an ace and the score > 21 then use ace as 1
4580 : IF SC%>21 AND ACE%>0 THEN SC% = SC%-10 : ACE% = ACE%-1
4590 UNTIL SC%<22 OR ACE%=0
4599 REM ** Check for bust
4600 IF SC%>21 THEN BUST = TRUE : SC% = 1
4610 RETURN
4998 REM ** Banker doubles **
4999 REM ** If card is a 10 or face card then only double sometimes
5000 IF VAL(HND$(0,1))=0 AND RND(1)>.6 THEN RETURN
5009 REM ** Inform players of doubling
5010 PING : WAIT 50 : PING
5020 MS$ = "          Banker doubles !" : GOSUB MESSAGE
5025 WAIT 200
5029 REM ** Double stakes
5030 FOR P=1 TO NP%
5039 REM ** Check for player still in
5040 : IF NOT IN(P) GOTO 5080
5049 REM ** Check whether player has enough credit to double
5050 : IF CR%(P)<2*ST%(P) THEN GOSUB CANT : GOTO 5080
5059 REM ** Double stake
5060 : ST%(P) = 2*ST%(P)
5070 : PLOT 10,PY%(P),MID$(STR$(ST%(P)),2)
5079 REM ** Display doubled stake by player's name
5080 NEXT P
5090 RETURN
5098 REM ** Player can't double **
5099 REM ** Inform player of his predicament
5100 MS$ = NM$(P)+" you can't double"
5105 GOSUB MESSAGE : WAIT 200
5110 MS$ = "You're out !"
5115 GOSUB MESSAGE
5119 REM ** Display 'OUT' where his stake was
5120 PLOT 10,PY%(P),"OUT"
5129 REM ** Remove player from the game
5130 IN(P) = FALSE
5140 WAIT 200
5150 RETURN
5198 REM ** Player has natural **
5199 REM ** Inform player he has a 'natural'
5200 MS$ = NM$(P)+" you have a natural !"
5210 GOSUB MESSAGE : PING
5219 REM ** Award bonus
5220 SC%(P) = 1000 : WAIT 200
5230 RETURN
5298 REM ** Illegal bet made **
5299 REM ** Inform player that his bet is illegal
5300 MUSIC 1,1,1,0 : PLAY 1,0,1,800
5309 REM ** Check if bet exceeds credit
5310 IF ST%>CR%(P) THEN MS$ = "You only have £"+MID$(STR$(CR%(P)),2)
5319 REM ** Check if bet is stupid !
5320 IF ST%<1 THEN MS$ = "Ok, joker ! Try again."
5340 GOSUB MESSAGE
5350 WAIT 200
5359 REM ** Set flag
5360 LEGAL = FALSE
5370 RETURN
5498 REM ** Print message in reserved area
5499 REM ** Clear bottom three lines
5500 FOR L=24 TO 26
5510 : PLOT 1,L,SP$ : PLOT 8,L,SP$
5520 NEXT L
5529 REM ** Set cursor to 25th line
5530 POKE #268,25
5539 REM ** Print message
5540 PRINT : PRINT MS$;
5550 RETURN
5598 REM ** House limit **
5599 REM ** Clear bottom three lines
5600 FOR L=24 TO 26
5610 : PLOT 1,L,SP$ : PLOT 7,L,SP$
5620 NEXT L
5629 REM ** Set cursor to 25th line
5630 POKE #268,25
5639 REM ** Display message
5640 PRINT : PRINT "        House limit = £499"
5650 MUSIC 1,1,1,0 : PLAY 1,0,1,800
5660 WAIT 200
5669 REM ** Set flag
5670 LEGAL = FALSE
5680 GOTO 1100
5681 REM ** Nb. GOSUB not used because it would exceed max. nesting
5698 REM ** Player's credit exhausted **
5699 REM ** Remove player from the game
5700 IN(P) = FALSE
5709 REM ** Inform player of his demise
5710 MUSIC 1,1,1,0 : PLAY 1,0,1,3000
5720 MS$ = NM$(P)+" is broke, goodbye !"
5730 GOSUB MESSAGE
5740 WAIT 200
5750 RETURN
5999 REM **  Shuffle cards **
6000 MS$ = "Shuffling, please wait." : GOSUB MESSAGE
6009 REM ** Shuffle cards by repeatedly picking two and swapping them
6010 FOR C=1 TO 104
6020 : S0% = RND(1)*52+1 : S1% = RND(1)*52+1
6030 : T$ = CARD$(S0%) : CARD$(S0%) = CARD$(S1%) : CARD$(S1%) = T$
6040 NEXT C
6050 NC% =-1
6060 RETURN
6498 REM ** Deal a card **
6499 REM ** Assign card to player P's hand and increment next card pointer
6500 HND$(P,CARD%) = CARD$(NC%) : NC% = NC%+1
6509 REM ** Check for end of pack
6510 IF NC%>52 THEN GOSUB 6000
6518 REM ** Display card
6519 REM ** Don't display banker's first two cards
6520 IF P=0 AND CARD%<3 THEN PLOT PX%(P,CARD%),PY%(P),"  ???"
6530 IF P>0 OR CARD%>2 THEN PLOT PX%(P,CARD%),PY%(P),MID$(HND$(P,CARD%),2)
6540 RETURN
6998 REM ** Prepare variables and screen **
6999 REM ** Clear stakes and cards
7000 FOR P=0 TO NP%
7010 : PLOT 9,PY%(P),SP$
7019 REM ** Empty hands
7020 : FOR C=1 TO 5
7030 :   HND$(P,C) = ""
7040 : NEXT C
7049 REM ** Reset stakes
7050 : ST%(P) = 0
7060 NEXT P
7070 RETURN
7999 REM ** Display board **
8000 CLS             'Clear screen
8009 REM ** Display top line
8010 FOR I=1 TO 40
8020 : POKE #BB7F+I,ASC(MID$(TITLE$,I,1))
8030 NEXT I
8039 REM ** Display headings in magenta (colour 5)
8040 PLOT 16,1,"BANKER" : PLOT 16,7,"PLAYERS" : PLOT 15,1,5 : PLOT 15,7,5
8049 REM ** Display separating lines (lines of ='s)
8050 PLOT 1,5,EQ$ : PLOT 1,23,EQ$ : PLOT 0,5,5 : PLOT 0,23,5
~~8060 PLOT 1,9," NAME   STAKE   CARDS" : PLOT 0,9,5~~
8069 REM ** Display players' data
8070 FOR PP=0 TO NP%
8079 REM ** Display name
8080 : PLOT 1,PY%(PP),NM$(PP)
8089 REM ** If player is out the display 'OUT' where stake would be
8090 : IF NOT IN(PP) THEN PLOT 10,PY%(PP),"OUT" : GOTO 8140
8099 REM ** If there is a current stake then display it
8100 : IF ST%(PP)>0 THEN PLOT 10,PY%(PP),MID$(STR$(ST%(PP)),2)
8109 REM ** Display cards
8110 : FOR C=1 TO 5
8120 :   PLOT PX%(PP,C),PY%(PP),MID$(HND$(PP,C),2)
8130 : NEXT C
8140 NEXT PP
8150 RETURN
8999 REM ** Set up game **
9000 CLS : PRINT : PRINT : PRINT
9009 REM ** Ask for number of players
9010 PRINT "How many players (max. = 6) ? ";
9019 REM ** Get number of players, only accept the numbers 1 to 6
9020 REPEAT
9030 : GET A$ : NP% = VAL(A$)
9040 UNTIL NP%>0 AND NP%<7
9045 PRINT A$ : PRINT : PRINT : PRINT
9049 REM ** Get each player's name
9050 FOR P=1 TO NP%
9060 : PRINT : PRINT "Player "P"'s name is ";
9069 REM ** If name is longer than 8 characters then just use first 8
9070 : INPUT NM$ : NM$(P) = LEFT$(NM$,8)
9080 NEXT P
9089 REM ** Player 0 is the computer
9090 NM$(0) = "COMPUTER"
9099 REM ** Set credit, stake and log player in
9100 FOR P=0 TO NP%
9110 : CR%(P) = 100 : ST%(P) = 0 : IN(P) = TRUE
9119 REM ** Empty hand
9120 : FOR C=1 TO 5
9130 :   HND$(P,C) = ""
9140 : NEXT C
9150 NEXT P
9159 REM ** Set computer's credit to max. (£32768)
9160 CR%(0) = #7FFF
9170 GOSUB BOARD
9180 RETURN
9998 REM ** Preliminaries **
9999 REM ** Clear screen and make it blue/white
10000 CLS : PAPER 7 : INK 4
10009 REM ** Dimension arrays
10010 DIM CARD$(52),HND$(6,5),NM$(6),S$(13)
10020 DIM IN(6),SC%(6),ST%(6),CR%(6),PY%(6),PX%(6,5)
10029 REM ** Define characters
10030 FOR I=1 TO 8
10040 : READ C
10050 : FOR J=0 TO 7
10060 :   READ D : POKE #B400+8*C+J,D
10070 : NEXT J
10080 NEXT I
10090 DATA 64,1,7,15,15,13,1,3,7       'Left half of spade
10091 DATA 91,0,48,56,56,24,0,32,48    'Right half of spade
10092 DATA 92,3,7,3,29,63,29,1,3       'Left club
10093 DATA 93,32,48,32,28,62,28,0,32   'Right club
10094 DATA 94,12,30,31,31,15,7,3,1     'Left heart
10095 DATA 123,24,60,60,60,56,48,32,0 'Right heart
10096 DATA 124,1,3,7,15,15,7,3,1       'Left diamond
10097 DATA 125,0,32,48,56,56,48,32,0   'Right diamond
10098 REM ** Build cards
~~10099 REM ** S$() = list of values~~
10100 FOR I=1 TO 13 : READ S$(I) : NEXT I
10105 DATA 1 A,2 2,3 3,4 4,5 5,6 6,7 7
10106 DATA 8 8,9 9,010,0 J,0 Q,0 K
10109 REM ** For each value
10110 FOR I=1 TO 13
10119 REM ** Spades
10120 : CARD$(I) = LEFT$(S$(I),1)+CHR$(0)+RIGHT$(S$(I),2)+"@["
10129 REM ** Clubs
10130 : CARD$(I+13) = LEFT$(S$(I),1)+CHR$(0)+RIGHT$(S$(I),2)+"\]"
10139 REM ** Hearts
10140 : CARD$(I+26) = LEFT$(S$(I),1)+CHR$(1)+RIGHT$(S$(I),2)+"^{"
10149 REM ** Diamonds
10150 : CARD$(I+39) = LEFT$(S$(I),1)+CHR$(1)+RIGHT$(S$(I),2)+"|}"
10160 NEXT I
10164 REM ** Cards are 6 characters long as follows:
10165 REM ** Chars    Meaning
10166 REM **    1     value for scoring routine
10167 REM **    2     colour: 0 = black, 1 = red
10168 REM **  3-4     value for display
10169 REM **  5-6     suit symbol
10199 REM ** Read in co-ords for plotting each players cards
10200 FOR P=0 TO 6
10210 : READ PY%(P)
10220 NEXT P
10225 DATA 3,11,13,15,17,19,21
10230 FOR C=1 TO 5
10240 : READ X%
10250 : FOR P=0 TO 6 : PX%(P,C) = X% : NEXT P
10260 NEXT C
10265 DATA 14,19,24,29,34
10299 REM ** Define line-numbers
10300 STAKE = 1000 : TWIST = 2000 : BANKER = 3000
10310 PAYOUT = 4000 : COUNT = 4500: DOUBLE = 5000
10320 CANT = 5100 : NATURAL = 5200 : BETBAD = 5300
10330 MESSAGE = 5500 : HOUSELIMIT = 5600 :BROKE = 5700
10340 SHUFFLE = 6000 : DEAL = 6500 :PREPARE = 7000
10341 REM ** This is not necessary but it does make reading the program easier
10350 BOARD = 8000 : SETUP = 9000
10398 REM ** Define strings
10399 REM ** 17 spaces, PONTOON, 16 spaces
10400 TITLE$="                 PONTOON                "
10409 REM ** 30 spaces
10410 SP$ = "                              "
10419 REM ** 36 ='s
10420 EQ$ = "===================================="
10499 REM ** Instructions
10500 PRINT : PRINT : PRINT
10510 PRINT "  This is the game of PONTOON" : PRINT : PRINT
10520 PRINT "  To play you try to get a hand of" : PRINT
10530 PRINT "cards which totals as close to 21 as" : PRINT
10540 PRINT "possible, aces count as either 1 or" : PRINT
10550 PRINT "11 as you wish and all face cards" : PRINT
10560 PRINT "count as ten." : PRINT : PRINT
10570 PRINT "  You are playing against the bank" : PRINT
10580 PRINT "and must beat it to win." : PRINT : PRINT
10590 PRINT "  The house limit on bets is _499."
10600 PRINT : PRINT
10610 PRINT "     Press SPACE to continue.";
10619 REM ** Wait until SPACE is pressed
10620 REPEAT : GET A$ : UNTIL A$ = " "
10900 RETURN
```

David Nowotnik

MACHINE CODE

Whether you're already writing machine code or just learning, this disassembler will make life easier.

Many programmers of home computers move onto machine code once they have acquired a good understanding of BASIC.

This trend seems to be motivated by speed; machine code programs can be more than a hundred times faster than their BASIC equivalents. But while programs are faster, programming can take very much longer.

In machine code you are supplying the CPU with a succession of simple commands in the form of numbers.

Often quite a few machine code commands are required to do the equivalent of one or two lines of BASIC. And while the list of numbers means a lot to the CPU, they usually mean little to the human operator.

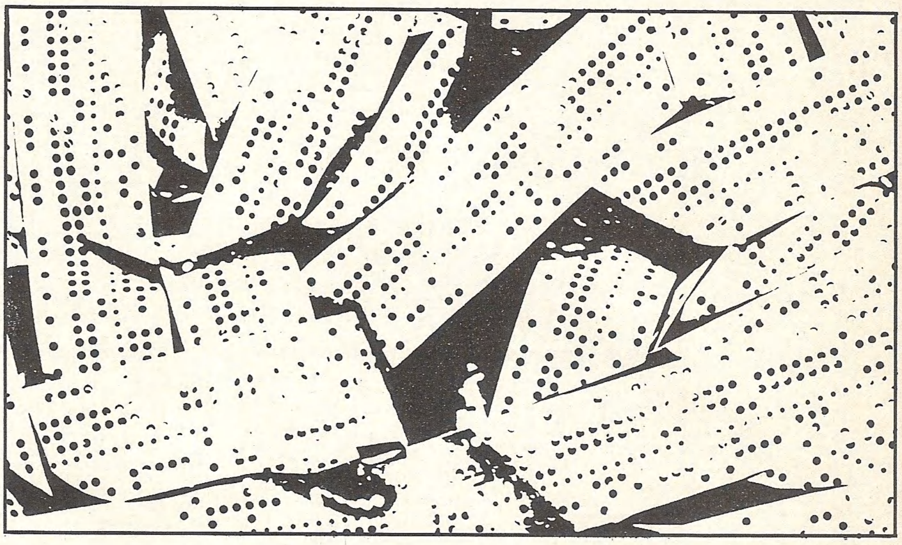
To help machine language programmers, an intermediate language was devised called assembly language. This is a translation of those numbers into a code more readily understood by man.

The process of converting machine code to assembly language is called disassembly, and this article presents a program for carrying out this process on the Oric-1.

If you know little about 6502 machine language programming, then I suggest you read one of the many good text books on the subject. If you already know a little, and want to learn more, dissambling other people's machine code routines can be an excellent way of helping you learn, and my disassembler program should help.

The menu-driven program (see **figures 1 and 2)** has several features.

It presents a disassembled listing of machine code on your TV screen or on the printer. If



you are writing machine code, then the program includes a hex loader to enter machine code through the keyboard, edited as required, and finally SAVEd on tape.

When you RUN the rogram, a title page is presented, followed by a short pause while various arrays are filled. The main menu, shown in **figure 1**, will appear. Two of the three main options will each give secondary menus, as shown in **figure 2.**

The SAVE/LOAD machine code option in the main menu gives further selectons.

Unfortunately, the first of these (load machine code from tape) will not work. A ROM error (one of many in the Oric) prevents CLOAD from working within a program. I've retained the option in the menu to present an apologetic message on the screen to serve as a reminder of Oric's failing.

If you wish to load machine code from tape, stop the program (option 4 in the main menu), and use CLOAD as a direct command to load the machine code, followed by RUN.

You can, however, SAVE machine code from within a program, and option 2 allows you to do this. After selection of this option, you will be asked for a file name, and the start and end addresses. Once the block of bytes have been saved, the SAVE/LOAD menu will re-appear.

Option 2 of the SAVE/LOAD menu allows you to enter your own machine code routine with a hex loader. First, you have to set the start address for your routine.

If you have a short routine, it can be stored in the safe area starting at address hex 400, otherwise it should go above HIMEM, which was set to hex 8000 by the program.

Enter your machine code as hexadecimal numbers. You should enter 'S' to stop; if you make a mistake, erase the error by pressing the delete key, and continue entering the numbers.

As you enter the numbers they will automatically be arranged on the screen in a hex dump array. Don't worry if you make a mistake and fail to spot it; you can use the edit routine to make any corrections.

Selection 2 from the main menu allows you to disassemble any routine you have placed in RAM, or any in-built routine in ROM. You start by specifying the start address, and whether you wish the output to be directed to the screen or printer.

Your machine code routine is disassembled one screenful at a time. On the screen (or printer) will be given the opcode address, the opcode mnemonic, the mode, the value of any operand, and the byte values of the entire machine code instruction.

While the opcode mnemonics are standard, the symbols for the mode are not. The system for display of modes is shown in **figure 3.** The abbreviations shown are directly attached to the three letter mnemonic of the opcode.

I find this system easier to understand than the standard method, hence my reason for using it in this program.

1 Save/Load machine code.
2 Disassemble machine code.
3 Edit machine code.
4 Stop.

Figure 1. Main menu

SAVE/ LOAD MACHINE CODE

1 Load machine code from tape.
2 Save machine code on tape.
3 Use hex loader.
4 Return to main menu.

EDIT MACHINE CODE

1 Alter a byte value.
2 Move a block of bytes.
3 Return to main menu.

Figure 2. Secondary menus

Once the screen is filled with a disassembled listing, pressing any key will give you the next 'page'. To stop the process, press 'S' at any time.

Editing of machine code in RAM is possible using the third option in the main menu. You may alter the value of a single byte, or move a block of bytes. Some caution has to be used with this option not to accidentaly alter the BASIC program, variables, or values in the first three pages of RAM.

If you want a short machine code routine to try out with the program, enter the listing in **figure 4** using the hex loader. Use 400 hex as the start address, and use CALL #400 as a direct command to find out what it does. The first 12 bytes are machine code, the rest is DATA.

Mode	Abbreviations
Implied	none
Accumulator	A
Absolute	none
Zero page	Z
Immediate	IM
Absolute, X	X
Absolute, Y	Y
Zero page, X	ZX
Zero page, Y	ZY
(Indirect, X)	IX
(Indirect), Y	IY
Relative	none

Figure 3. Abbreviations for modes

```
A2 20 BD 0B 04 9D 84 BB CA
D0 F7 60 11 52 65 6D 65 6D 62
65 72 20 74 6F 20 67 65 74 20
48 43 57 20 6E 65 78 74 20 77
65 65 6B 20 10
```

Figure 4. An example machine code routine to try

```
10 REM   6502 DISASSEMBLER
20 REM        David Nowotnik
30 REM
40 REM
50 REM
60 REM       48K version
70 HIMEM#7FFF:TEXT:PAPER0:INK7:PRINTCHR$
(17)
80 DIMOC$(256),OP(256),SC$(400)
90 E$=" "+CHR$(27)
100 LS$=" ##############################
######"
105 POKE618,2:IFPEEK(524)<128THENPRINTCH
R$(20);
110 HM$=CHR$(30):D$=CHR$(10):U$=CHR$(11)
:LT$=CHR$(8)
130 RT$="":FORI=1TO8:RT$=RT$+CHR$(9):NEX
T:DH$=CHR$(4)
140 GOSUB200:PRINTHM$D$D$D$D$D$RT$RT$DH$
E$"J6502"DH$
150 PRINTD$D$D$RT$"HEX LOADER/DISASSEMBL
ER"
160 PRINTD$D$D$RT$"  FOR THE 48K ORIC 1"
170 PRINTD$D$D$RT$"                1983, D.
 Nowotnik"
190 GOTO800
200 CLS:PRINT:PRINTLS$
210 FORI=1TO20:PRINT" #";SPC(34);"#"
220 NEXTI:PRINTLS$
230 RETURN
250 HA$=HEX$(W):HA$="0000"+HA$:HA$=RIGHT
$(HA$,4)
260 W$=STR$(W):W$="     "+W$:W$=RIGHT$(W
$,7):W$=HA$+W$
270 V$=HEX$(T):IFLEFT$(V$,1)="#"THENV$=R
IGHT$(V$,LEN(V$)-1)
280 V$=RIGHT$("00"+V$,2)
290 K$="  "+K$+"            ":K$=LEFT$(K
$,8):RETURN
300 B=PEEK(W+1):G$=HEX$(B):IFB>16THENV$=
V$+RIGHT$(G$,2):RETURN
305 V$=V$+"0"+RIGHT$(G$,1):RETURN
310 NM$=RIGHT$("            "+STR$(PEEK(
W+1)),10)+"   ":RETURN
320 PRINTD$" For a printer listing press
 'P'"
330 PRINTD$"   else press any other key"
340 GETP$:RETURN
350 PRINTD$D$"Enter file name ";:INPUTNF
$
360 RETURN
370 PRINTD$"Enter start address ";:INPUT
SA
380 PRINTD$"Enter end address ";:INPUTEA
```

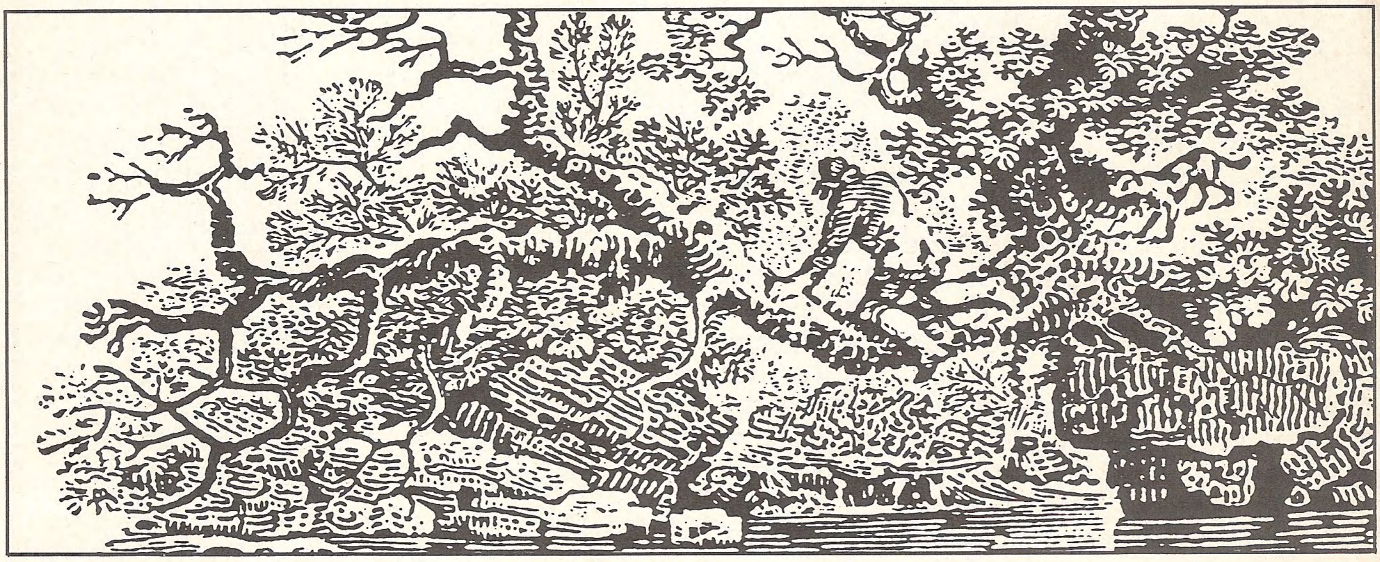
```
390 RETURN
400 PRINTD$D$" Start tape, press a key,
then wait"
410 GETZ$:WAIT600:RETURN
420 HI=ASC(LEFT$(S$,1))-48:IFLEFT$(S$,1)
>"9"THENHI=HI-7
430 LO=ASC(RIGHT$(S$,1))-48:IFRIGHT$(S$,
1)>"9"THENLO=LO-7
440 HI=HI*16:RETURN
700 TEXT:PAPER0:INK7:CLS
800 REM INITIALISE
810 FORI=1TO256
820 READOC$(I):IFOC$(I)="X"THEN840
830 READOP(I)
840 NEXTI
850 GOSUB200
860 PRINTHM$D$D$D$D$RT$RT$"MENU"
870 PRINTD$D$RT$"1.  Save/load machine
 code"
880 PRINTD$D$RT$"2.  Disassemble Machine
 code"
885 PRINTD$D$RT$"3.  Edit machine code"
890 PRINTD$D$RT$"4.  Stop"
895 PRINTD$D$RT$" *** Select 1 to 4 ***"
900 GETZ$:Z=VAL(Z$):IFZ<1ORZ>4THEN900
910 PRINTU$RT$"            ";Z;" Selected
 "
920 WAIT200
940 IFZ=4THENCLS:STOP
960 ONZGOSUB1000,2000,3000
970 GOTO850
1000 REM Load machine code routine
1010 GOSUB200:PRINTHM$D$D$D$RT$"  SAVE /
LOAD MACHINE CODE"
1020 PRINTD$RT$"         Options"
1030 PRINTD$D$D$RT$"1. Load machine code
 from tape"
1032 PRINTD$RT$"2. Save machine code on
tape"
1036 PRINTD$RT$"3. Use the hex loader"
1040 PRINTD$RT$"4. Return to the main me
nu"
1050 PRINTD$D$D$RT$"  *** Select 1 to 4
***"
1060 GETV$:U=VAL(V$):IFU<1ORU>4THEN1060
1070 PRINTU$RT$"       ";U;" selected
 "
1080 WAIT200:IFU=4THENRETURN
1090 CLS:ONUGOSUB1100,1300,1500:GOTO1000
1100 CLS:PRINTD$RT$"Load machine code fr
om tape"
1110 PRINTD$D$D$D$"     Due to ROM fault
, cannot do."
1120 PRINTD$D$"                Sorry!":
WAIT1000:RETURN
1130 CLOADNF$,A,E
1140 RETURN
1300 CLS:PRINTD$RT$"Save machine code on
 tape"
1310 GOSUB350:GOSUB370:GOSUB400
1320 CSAVENF$,A(SA),E(EA)
1330 RETURN
1500 CLS:PRINTD$RT$"Hex loader"
1510 PRINTD$D$"Enter the start address "
;:INPUTSA:S1=SA
1520 PRINTD$D$"Enter machine code as hex
 digit pairs,";
1530 PRINT"and 'S' to stop and return to
 the mainmenu."
1540 PRINT
1550 Z$=RIGHT$("0000"+HEX$(SA),4)
1560 PRINTZ$;"     ";:MN=0
1570 S$="":FORI=1TO2
1580 GETZ$:IFZ$="S"THENWAIT400:RETURN
1585 IFASC(Z$)=127THENGOSUB1700:GOTO1580
1590 IFZ$<"0"ORZ$>"F"THEN1580
1600 IFZ$>"9"ANDZ$<"A"THEN1580
1610 PRINTZ$;:S$=S$+Z$:NEXTI
1620 PRINT"  ";:GOSUB420:POKESA,HI+LO
1630 SA=SA+1:MN=MN+1:IFMN<8THEN1570
1640 GOTO1540
1700 IFI=2THENI=I-1:S$="":PRINTLT$" "LT$
;:RETURN
1710 IFMN=0THEN1725
1720 PRINTLT$LT$LT$LT$"  "LT$LT$;:MN=MN-
1:SA=SA-1:RETURN
1725 IFSA=S1THENRETURN
1730 PRINT:PRINTU$U$LT$LT$"  "LT$LT$;
1740 SA=SA-1:MN=7:RETURN
2000 REM Disassemble routine
2010 CLS:PRINTD$RT$"  6502 DISASSEMBLE R
OUTINE"
2020 PRINTD$:INPUT"  Enter the start add
ress  ";W
2030 IFW<0ORW>65535THENPRINTD$D$"  Addre
ss out of range":WAIT400:GO
TO2000
2040 IFW<>INT(W)THENPRINTD$D$"   Integer
 values only:WAIT400:GOTO20
00
2045 GOSUB320
2050 CLS:L1=0:PRINT" Address  Opcode  Op
erand  Byte value8"
2060 PRINT:IFP$="P"THENLPRINT"     Addre
ss      Opcode
  Operand"
2070 T=PEEK(W):K$=OC$(T+1):K=OP(T+1)
2080 GOSUB250
2090 IFK>0THEN2140
2100 PRINTW$;"  Invalid Opcode    ";V$
2110 IFP$="P"THENLPRINTW$;TAB(10);" Inva.
lid Operand ;V$
2120 W=W+1:L1=L1+1:IFL1>21THEN2500
2130 GOTO2070
2140 ONKGOTO2150,2200,2300
2150 PRINTW$K$"            "V$
```

```
2160 IFP$="P"THENLPRINTW$K$"
"V$
2170 GOTO2120
2200 MK=0:GOSUB300:FORX=16TO240STEP32
2210 IFX<>T1HEN2250
2220 MK=1:JR=PEEK(W+1)+2:IFJR>129THEN224
0
2230 JR$=HEX$(W+JR):JR$=RIGHT$("0000"+JR
$,4):GOTO2280
2240 JR$=HEX$(W-(256-JR)):JR$=RIGHT$("00
00"+JR$,4):GOTO2280
2250 NEXTX:GOSUB310
2260 IFP$="P"THENLPRINTW$K$NM$V$
2270 PRINTW$K$NM$V$:W=W+2:L1=L1+1:IFL1>2
1THEN2500ELSEGOTO2070
2280 NM$="  to "+JR$+"     ":GOTO2260
2300 NM$=RIGHT$("        "+STR$(DEEK(W+1))
,10)+"   "
2310 GOSUB300:W=W+1:GOSUB300:W=W-1
2320 IFP$="P"THENLPRINTW$K$NM$V$
2330 PRINTW$K$NM$V$
2340 W=W+3:L1=L1+1:IFL1>23THEN2500ELSEGO
TO2070
2500 IFKEY$="S"THENWAIT200:RETURN
2510 IFP$="P"THEN2050
2520 GETZ$:IFZ$<>"S"THEN2050
2530 RETURN
3000 GOSUB200:PRINTHM$D$D$D$RT$"      Edi
t Machine Code"
3010 PRINTD$D$D$RT$"1.   Alter a byte va
lue"
3020 PRINTD$RT$"2.   Move a block of byt
es"
3030 PRINTD$RT$"3.   Return to the main
menu."
3040 PRINTD$D$D$D$D$RT$"     Select 1,
2,or 3"
3050 GETZ$:Z=VAL(Z$):IFZ<1ORZ>3THEN3050
3060 IFZ=3THENRETURN
3070 IFZ=2THEN3500
3080 CLS:PRINTD$RT$"        Alter a byte
 value"
3090 PRINTD$D$"   Input byte address ";:
INPUTBA
3100 PRINTD$D$"  Present value = ";PEEK(
BA)
3110 PRINTD$D$" Enter new value  ";:INPU
TNV
3120 IFNV<0ORNV>255ORNV<>INT(NV)THEN3110
3130 POKEBA,NV
3140 PRINTD$D$RT$"Any more (Y/N)"
3150 GETZ$:IFZ$="Y"THEN3080
3160 IFZ$="N"THEN3000
3170 GOTO3150
3500 CLS:PRINTD$RT$"Move a block of byte
s"
3510 PRINTD$D$D$"Enter present start add
ress ";:INPUTSA
3520 PRINTD$D$"Enter new start address "
;:INPUTNA
3530 PRINTD$D$"Enter number of bytes ";:
INPUTBY
3540 PRINTD$D$RT$"      Please wait"
3550 IFSA=NATHEN3000
3560 IFSA>NATHEN3700
3570 NA=NA+BY-1
3580 FORI=SA+BY-1TOSASTEP-1
3590 POKENA,PEEK(I):NA=NA-1
3600 NEXTI:GOTO3000
3700 FORI=SATOSA+BY-1
3710 POKENA,PEEK(I):NA=NA+1
3720 NEXTI:GOTO3000
5000 DATABRK,1,ORAIX,2,X,X,X,ORAZ,2
5010 DATAASLZ,2,X,PHP,1,ORAIM,2
5020 DATAASLA,1,X,X,ORA,3,ASL,3,X
5030 DATABPL,2,ORAIY,2,X,X,X,ORAZX,2
5040 DATAASLZX,2,X,CLC,1,ORAY,3,X,X,X
5050 DATAORAX,3,ASLX,3,X,JSR,3,ANDIX,2
5060 DATAX,X,BITZ,2,ANDZ,2,ROLZ,2,X
5070 DATAPLP,1,ANDIM,2,ROLA,1,X,BIT,3
5080 DATAAND,3,ROL,3,X,BMI,2,ANDIY,2
5090 DATAX,X,X,ANDZX,2,ROLZX,2,X,SEC,1
5100 DATAANDY,3,X,X,X,ANDX,3,ROLX,3,X
5110 DATARTI,1,EORIX,2,X,X,X,EORZ,2
5120 DATALSRZ,2,X,PHA,1,EORIM,2,LSRA,1
5130 DATAX,JMP,3,EOR,3,LSR,3,X,BVC,2
5140 DATAEORIY,2,X,X,X,EORZX,2,LSRZX,2
5150 DATAX,CLI,1,EORY,3,X,X,X,EORX,3
5160 DATALSRX,3,X,RTS,1,ADCIX,2,X,X,X
5170 DATAADCZ,2,RORZ,2,X,PLA,1,ADCIM,2
5180 DATARORA,1,X,JMPI,3,ADC,3,ROR,3
5190 DATAX,BVS,2,ADCIY,2,X,X,X,ADCZX,2
5200 DATARORZX,2,X,SEI,1,ADCY,3,X,X,X
5210 DATAADCX,3,RORX,3,X,X,STAIX,2
5220 DATAX,X,STYZ,2,STAZ,2,STXZ,2,X
5230 DATADEY,1,X,TXA,1,X,STY,3,STA,3
5240 DATASTX,3,X,BCC,2,STAIY,2,X,X
5250 DATASTYZX,2,STAZX,2,STXZY,2,X
5260 DATATYA,1,STAY,3,TXS,1,X,X,STAX,3
5270 DATAX,X,LDYIM,2,LDAIX,2,LDXIM,2
5280 DATAX,LDYZ,2,LDAZ,2,LDXZ,2,X
5290 DATATAY,1,LADIM,2,TAX,1,X,LDY,3
5300 DATALDA,3,LDX,3,X,BCS,2,LDAIY,2
5310 DATAX,X,LDYZX,2,LDAZX,2,LDXZY,2
5320 DATAX,CLV,1,LDAY,3,TSX,1,X,LDYX,3
5330 DATALDAX,3,LDXY,3,X,CPYIM,0
5340 DATACMPIX,2,X,X,CPYZ,2,CMPZ,2
5350 DATADECZ,2,X,INY,1,CMPIM,2,DEX,1
5360 DATAX,CPY,3,CMP,3,DEC,3,X,BNE,2
5370 DATACMPIY,2,X,X,X,CMPZX,2,DECZX,2
5380 DATAX,CLD,1,CMPY,3,X,X,X,CMPX,3
5390 DATADECX,3,X,CPXIM,2,SBCIX,2,X,X
5400 DATACPXZ,2,SBCZ,2,INCZ,2,X,INX,1
5410 DATASBCIM,2,NOP,1,X,CPX,3,SBC,3
5420 DATAINC,3,X,BEQ,2,SBCIY,2,X,X,X
5430 DATASBCZX,2,INCZX,2,X,SED,1
5440 DATASBCY,3,X,X,X,SBCX,3,INCX,3,X
```

S W Lucas

FOREST TREK

Playing this game may be a case of not seeing the wood for the trees! All you have to do is guide Fred through the forest.



Forest Trek is a game for the Oric/Atmos 48K. It will also work in the 16K machine if the POKE addresses are changed by subtracting 8000 hex. Although the game's listing is only short, it is fun to play, and each screen that you clear cause the game to get harder. I chose not to add too many sound effects because early Oric computers have a habit of causing screen wobble (on TV sets not monitors) when PING, SHOOT, EXPLODE and ZAP are used. If you want more effects just add line 205 PING.

You must guide Fred through the forest to the hut at the top right of the screen using the following keys:

Z — left
X — right
; — up
. — down

You must avoid bumping into trees and spiders!

Each time you have completed a screen, you score one point and the game gets harder because the spiders appear more often!

RUNdown

Line	Action
3	Calls subroutine for instructions.
6	POKEs disable the keyboard noise.
8	Chooses text mode and appropriate background and foreground colours.
9-60	Redefines the character set . . . users with 16K machines will need to alter the addresses in the POKE statements by subtracting 8000 hex. I have chosen the redefine several characters such as ?/£% & which are not otherwise used.
70-85	Draws the blue border round the screen.
90-110	Plots 50 trees in random positions on the screen.
115	Defines the characters available in the game.
120-135	Plots the start and end positions on the screen.
140	Defines start position of 'FRED'.
146	Prints score on screen.
150-300	This is the main control loop of the game which repeats the process of waiting for key to be pressed and moving the character until FRED hits another object.
152-154	Plots a spider at a random position after a time delay set by the choice of the variable R.
155	Remembers current position.
160	Key $ equivalent to inkey$ on most machines.
165-190	Chooses new position of FRED.
192	Looks to see if position is occupied.
198	Checks to see the character that is at the new location.
200	Moves FRED.
300	End of main loop.
310	Checks to see if you win or lose!

390	Increment score and decrease delay before spider appears.
400-450	Gives score and waits for space bar.
1000-1090	Lose game messages.
2000-2180	Subroutine for instructions.
10000-	Data to redefine character set.

VARIABLES USED

X	Current horizontal screen coordinate.
Y	Current vertical screen coordinate.
XX	Previous horizontal coordinate.
YY	Previous vertical coordinate.
S	Score.
X and Y	Used in counting loops for POKEing data into change the character set.
R	Delay before spiders appear.
T	Counting loop to count until spider appears.
U and V	Hold random numbers of the position of the spiders on the screen.
AA$	Uses the key$ function to get a character from the keyboard.
Z	Test to see if you hit object.
P and Q	Holds the ASCII value of the character at location X,Y on the screen.

CONVERSION CLUES

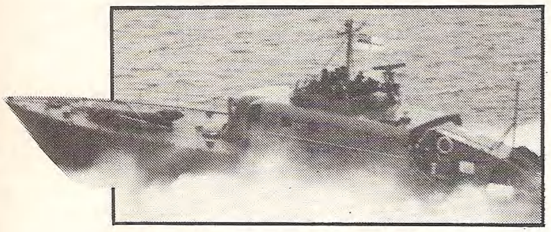
1. You would need to define an appropriate character set for the machine you are using.
2. The Oric uses serial attributes, which means that the colour changes occupy character spaces on the screen. For this reason, I have kept the majority of characters on the screen in red (serial attribute 1) or blue (serial attribute 4).
3. Most machines use inkey$ rather than key$.
4. SCRN(x,y) returns the ASCII code of the character found at location x,y on the text screen. The origin (0,0) being the top left of the screen, with the maximum values being (38,26).

```
1 REM :: FOREST TREK :  a game for the 48 K Oric 1 and Atmos computers from
2 LET Z=0
3 GOSUB2000
5 REM :: S. W. Lucas   February 1984 ::
6 POKE£26A,10:POKE524,255
8 TEXT:GRAB:PAPER7:INK4:CLS
9 REM :: define character set ::
10 FORX=46400TO46431:READY:POKEX,Y:NEXT
11 S=0
20 FORX=46360TO46391:READY:POKEX,Y:NEXT
30 FORX=46560TO46575:READY:POKEX,Y:NEXT
40 FORX=46344TO46351:READY:POKEX,Y:NEXT
50 FORX=46456TO46463:READY:POKEX,Y:NEXT
60 FORX=46576TO46591:READY:POKEX,Y:NEXT
65 LET R=25
70 FORX=1TO37:PLOTX,1,"?":PLOTX,25,"?":NEXT
80 FORX=2TO24:PLOT1,X,"?":PLOT37,X,"?":NEXT
85 FORX=2TO23:PLOT2,X,1:PLOT36,X,4:NEXT
90 FORT=1TO50
91 X=INT(RND(1)*31)+3:Y=INT(RND(1)*20)+3
92 IF SCRN(X,Y)<>32ORSCRN(X+1,Y)<>32 THEN91
100 PLOTX,Y,"£$"
110 NEXT
111 T=0
115 A$="%&":B$="£$":C$="()":D$="*+":E$="!/":F$=">?":G$="<="
120 PLOT33,3,5:PLOT34,3,E$:PLOT36,3,4:PLOT30,0,0:PLOT31,0,"Finish"
130 PLOT1,26,0:PLOT3,26,"Start":PLOT4,25,"  ":PLOT3,24,5:PLOT4,24,C$
135 PLOT6,24,1:PLOT35,24,4
140 LET X=6:LET Y=23
146 PRINT"SCORE ;- ";S
150 REPEAT
151 T=T+1
152 IFT>RTHEN U=INT(RND(1)*31)+3:V=INT(RND(1)*20)+4
153 IFT>RAND(U=XORV=Y)THEN152
154 IFT>RTHENPLOTU,V,A$:T=0:ZAP
155 XX=X:YY=Y
160 AA$=KEY$
165 IFAA$="Z"THEN X=X-1:IFX<3THENX=3
170 IFAA$="X"THEN X=X+1:IFX>34THENX=34
180 IFAA$="."THEN Y=Y+1:IFY>23 THEN Y=23
190 IFAA$=";"THEN Y=Y-1:IFY<2 THENY=2
192 P=SCRN(X,Y):Q=SCRN(X+1,Y)
193 IF(P=32ORP=42ORP=43)AND(Q=32ORQ=42ORQ=43)THENZ=0 ELSE Z=1
200 PLOTXX,YY,"  ":PLOTX,Y,D$
300 UNTIL Z=1
310 IFP=37ORP=38ORP=35ORP=36ORQ=37ORQ=38ORQ=35ORQ=36THENEXPLODE:GOTO1000
390 CLS:PRINT"WELL DONE ... ":LET S=S+1:LET R=R-2:IFR<0THENLETR=0
400 PRINT:PRINT:PRINT"You have now scored :-";S
410 PRINT:PRINT:PRINT"PRESS  SPACE BAR FOR NEXT SHEET"
420 REPEAT
430 GETA$
440 UNTIL A$=" "
445 CLS
450 GOTO70
500 END
1000 CLS:INK1:PRINT:PRINT:PRINT:PRINT
1010 PRINT"Oh dear you have just got yourself    killed !/"
1015 PRINT:PRINT"You scored :-";S
1020 PRINT:PRINT:PRINT:PRINT"Would you like another game   Y N   "
1030 REPEAT
1040 GET A$
1050 UNTIL A$="Y" OR A$="N"
1060 IF A$="Y" THENPRINTCHR$(17):RUN
1070 POKE £26A,3:CLS
1080 PLOT3,10,"Goodbye... thank you for playing"
1090 END
2000 CLS:PAPER0:INK3
2010 FORX=1TO2:PLOT1,X,X:PLOT2,X,14:PLOT12,X,"Forest trek":NEXT
2015 PRINT:PRINT:PRINT
2020 PRINT:PRINT:PRINT"A  game by  S.W. Lucas  1984"
2030 PRINT:PRINT:PRINT"Your task in this game is to guide"
2040 PRINT"Fred through the forest using the keys"
2050 PRINT:PRINT"  Z - LEFT   X - RIGHT"
2060 PRINT:PRINT"  ; - UP     . - DOWN"
2070 PRINT:PRINT"You must avoid the trees and the"
2080 PRINT"deadly spiders."
2090 PRINT"Each time you complete a sheet, the "
2100 PRINT"game gets harder, with more spiders to";
2110 PRINT"avoid.  You start at the bottom left"
2120 PRINT"of the screen and finish at the top "
2130 PRINT"right of the screen"
2140 PRINT:PRINT:PRINT"Press SPACE BAR to start game"
2150 REPEAT
2160 GETA$
2170 UNTIL A$=" "
2180 CLS:RETURN
10000 DATA1,2,4,9,16,21,17,0,32,16,8,36,2,42,34,0,1,1,15,11,3,2,6,0,32,32
10010 DATA60,52,48,16,24,0,1,3,7,15,31,1,3,0,0,32,48,56,60,0,32,0,0,0,4
10020 DATA3,15,10,11,0,0,0,4,56,62,42,26,0,0,0,15,15,15,8,8,0,0,8,60,44
10030 DATA36,32,32,0,40,56,56,63,44,60,47,0,0,0,0,62,50,50,62,0,63,34
10040 DATA63,34,63,34,62,0,63,17,63,17,63,17,31,0
```

M.R. Draper

ZAP 'EM

Can you save your country from the fleet of enemy ships?



Your country is threatened by ememy ships — you must use your skill as the commander of a missile battery to sink those ships as they cross the horizon before they have a chance to destroy you.

All instructions come up on screen as well as your scoring possibilities and final tally.

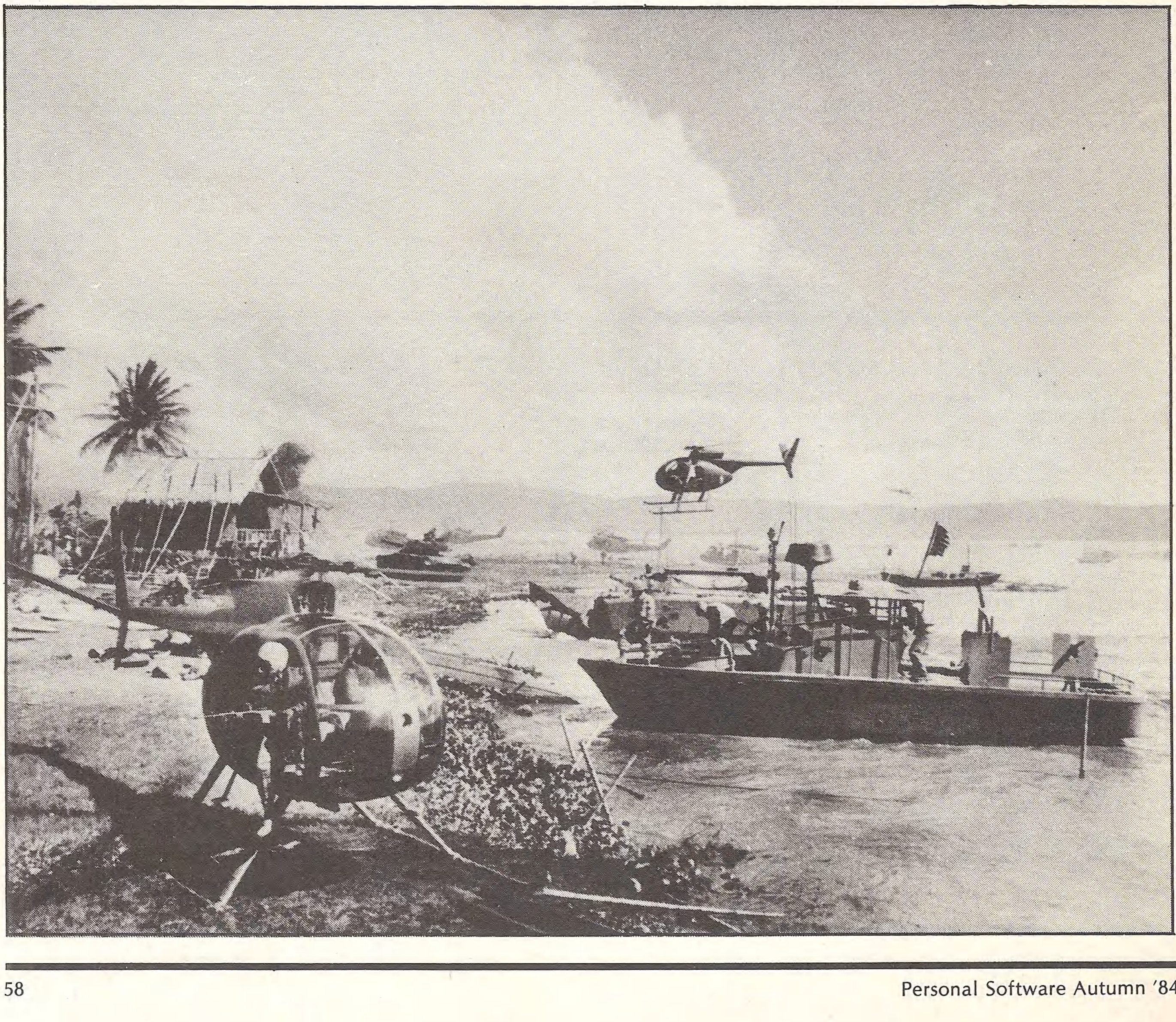
Score form 50 for an early hit to 10 for 'nearly missed'.

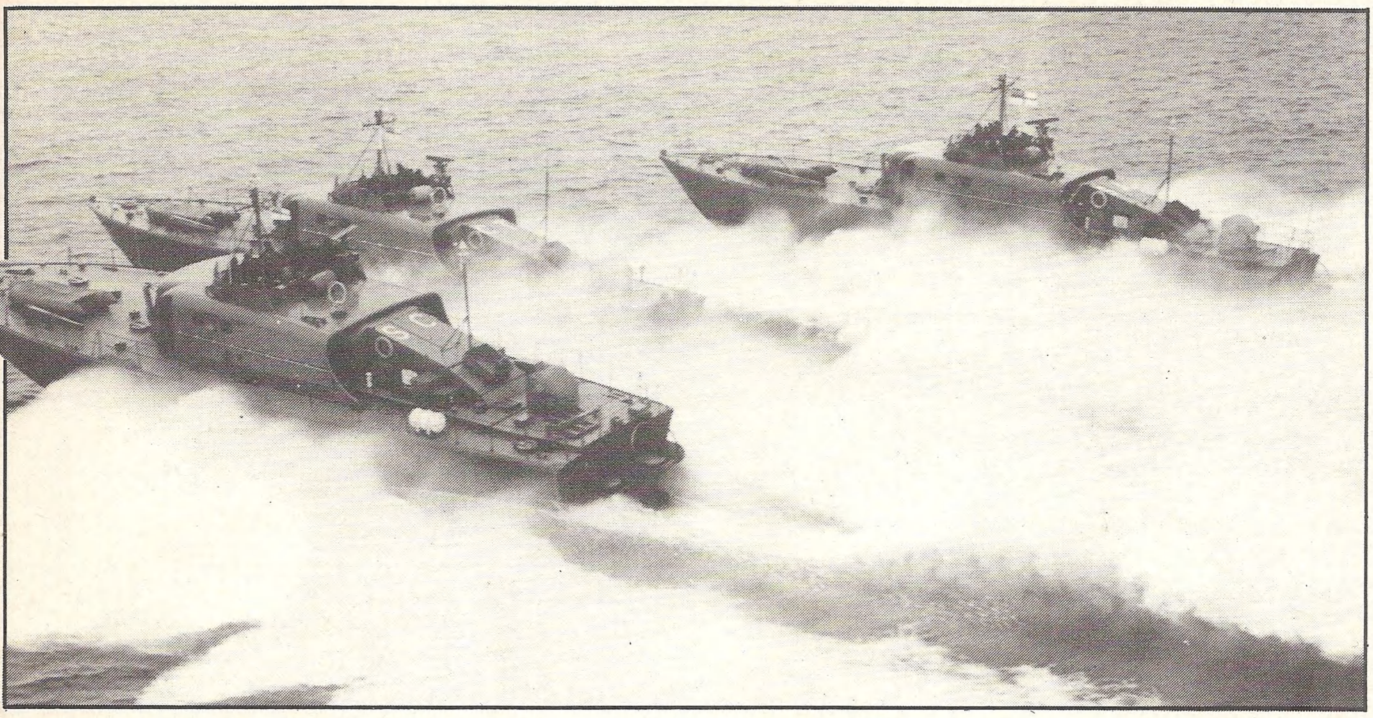
HOW IT WORKS

1-20	initialise program
30	data input
40	game instructions display and score display
50	define enemy ship patterns
60	define battlefield
70-470	main game loop

VARIABLES

COL and LIN	used in PLOT statements for positioning sight
S$	holds characters of the enemy ships
A$	RND statement to decide which ship crosses horizon
MI	number of missiles left
HS	high score
SC	present score
Q$	numbers for a little tune played at times during game.





```
~
1 REM MISSILE ATTACK
2 REM BY M.R.DRAPER
3 REM FOR THE ORIC-1
10 DIMS$(100)
20 POKE 618,10
30 GOSUB9000
40 GOSUB4000:INK7
50 GOSUB1000
60 GOSUB1170
70 HS=0
100 FORX=1TO100:J=0:AA$=S$(INT(RND(1)*3+1))
110 FORXZ=1TO32:WAIT5
120 PLOT0,5,"                                  "
130 PLOTXZ,5,AA$
140 PLOTCOL,LIN," "
150 K$="A":K$=KEY$
160 IFCOL<=4THENCOL=4
170 IFK$<>"Z"THEN190
180 COL=COL-2
190 IFCOL>=36THENCOL=36
200 IFK$<>"X"THEN220
210 COL=COL+2
220 PLOTCOL,LIN,"+"
230 PLOTCOL,4," "
240 IFK$<>"L"THEN310
250 L=0:Y=20
260 MI=MI-1
270 PLOTCOL-1,5,"g"
280 PLOTCOL+1,5,"h"
290 SHOOT:EXPLODE:GOSUB6000
300 IFJ=1THEN340
310 WAIT5
320 IFMI=<0THEN3000
330 NEXTXZ
340 NEXTX
350 GOTO3000
360 REM TUNE
370 Q$="14617356173561735"
380 FORQ=1TOLEN(Q$)
390 W=ASC(MID$(Q$,Q,1))-47
400 PLAY 7,0,5,2500
410 MUSIC1,1,W,10
420 MUSIC2,2,W,10
430 MUSIC3,3,W,10
440 WAIT 15
450 NEXT Q
460 PLAY 0,0,0,0
470 RETURN
1000 PAPER4:CLS:LIN=22:COL=15:MI=10:PRINT
1010 FORX=1TO5:PRINTCHR$(27);CHR$(85):NEXTX
1020 GOSUB6070
1030 RETURN
1170 S$(1)="a                         "
1180 S$(2)="bc                        "
1190 S$(3)="def                       "
1200 FORX=4TO6
1210 S$(X)="                          "
1220 NEXTX
1230 SC=0
1240 RETURN
3000 SC$=STR$(SC):PLOT6,10,"YOUR SCORE WAS "
3010 PLOT20,10,SC$:WAIT50
3020 SC=0:MI=10
3030 PLOT3,16,"PRESS ANY KEY TO PLAY"
3040 GOSUB360:GETA$
3050 GOSUB1000
3060 GOSUB1170
3070 GOTO100
4000 CLS:PAPER1:INK0
4001 PRINT:PRINT:PRINT:PRINT,,;
4002 POKE#26A,90:PRINT"J  MISSILE ATTACK"
4003 PRINTCHR$(4):PRINT:PRINT:PRINT
4004 PRINT" THE OBJECT OF THIS GAME IS TO":PRINT:PRINT
4005 PRINT" SINK THE SHIPS WITH YOUR MISSILES":PRINT:PR
INT
4006 PRINT" AS THEY CROSS THE HORIZON":GOSUB360:CLS:PRI
NT
4007 PRINT"* * * * * * * * * * * * * * * * * * *"
4008 PRINT:PRINT,,;"THE CONTROLS "
4009 PRINT:PRINT
4010 PRINT,,"Z--MOVE LEFT":PRINT
4011 PRINT,,"X--MOVE RIGHT":PRINT
4012 PRINT,,"L--TO FIRE":PRINT
4013 PRINT:PRINT"* * * * * * * * * * * * * * * * * * *"
4014 PRINT:PRINT,,;"SCORING"
4015 PRINT:PRINT
4016 PRINT,,"def   SCORES 10":PRINT
4017 PRINT,,"bc    SCORES 25":PRINT
4018 PRINT,,"a     SCORES 50":PRINT
4019 PRINT:PRINT"PRESS ANY KEY TO PLAY THE GAME"
4020 GOSUB360:RETURN
4050 GET A$
6000 REM
6010 IFSCRN(COL,5)=97THENSC=SC+50:J=1
6020 IFSCRN(COL,5)=98ORSCRN(COL,5)=99THENSC=SC+25:J=1
6030 IFSCRN(COL,5)=100ORSCRN(COL,5)=101ORSCRN(COL,5)=10
2THEN6050
6040 GOTO6060
6050 SC=SC+10:J=1
6060 IFSC>HSTHENHS=SC
6070 PLOT3,0,"SCORE:     MISSE:    HIGH SCORE:"
6080 SC$=STR$(SC)
6090 PLOT9,0,SC$
6100 MI$=STR$(MI):PLOT20,0,MI$
6110 HS$=STR$(HS):PLOT35,0,HS$
6120 RETURN
9000 RESTORE:FORU=46856TO46919
9010 READG
9020 POKEU,G
9030 NEXTU
9040 DATA0,0,0,0,0,0,16,255
9050 DATA0,0,0,0,1,39,255,127
9060 DATA0,0,0,0,128,230,255,254
9070 DATA0,0,0,0,1,255,127,63
9080 DATA0,24,60,63,255,255,255,255
9090 DATA0,0,0,0,192,255,254,252
9100 DATA8,4,2,63,2,4,8,16
9110 DATA16,32,64,63,64,32,16,8
9120 RETURN
```

Alasdair Jarvie

HELP AN ALIEN

Instead of zapping aliens, why not help this one back to his spaceship! This game, which runs on an Oric or Atmos, gives the chance to be kind to a being from outer space.

Alien is lost and cannot find his way back to the spaceship unless you help. You have to guide him up the ladders, avoiding the bombs.

You collect bonus points when you reach the spaceship or move underneath an object.

The game also has a "choose your own volume" feature so your family won't be bothered by the noise! It runs in 16K and will work on any Oric or Atmos.

VARIABLES

A,B	position of alien
C,D,E,F	position of spaceship
G,H,I,J,K,L	positions of bombs

HINTS ON CONVERSION

PLOT can be replaced by PRINT AT, PAPER and INK change the colour of the screen, SCRN checks what is on the screen, ABS is the absolute value and POKE 618, 10 switches off cursor and keyclick.

HOW IT WORKS

10-50	variables
60-90	get input and react
100-200, -300, -400, -500	move alien left, right, down, up
500-600	update score etc
600-900	move bomb 1, bomb 2, bomb 3
10000-19999	redefine characters
20000-29999	instructions
30000-39999	set up screen
40000-45000	end
50000-50000	move to another screen

```
 1 TEXT
 8 POKE 618,10
 10 REM *** ALIEN ***
 11 POKE 618,10
 20 REM *** BY A. JARVIE          ***
 30 LIVES=3:P=40000
 40 GOSUB 10000:GOSUB 55000:GOSUB 20000
 50 GOSUB 30000
 60 A=18:B=19:C=4:D=1:E=4:F=2:G=4:H=3:I=-34:J=7:K=4:L=1
1:Q$="de":W$="fg"
 70 PLOT A,B,"h":PLOTC,D,Q$:PLOTE,F,W$:PLOT ABS(G),H,"l
":PLOTABS(I),J,"l"
 75 PLOT ABS(K),L,"l"
 80 A$=KEY$
 85 IF A$="Z"THENGOTO100
 87 IF A$="X"THENGOTO200
 89 IF A$="/"THENGOTO300
 91 IF A$="'"THENGOTO400
 95 GOSUB500:GOSUB600:GOSUB700:GOSUB800
 97 GOTO 70
 100 REM *** ALIEN - LEFT ***
 105 IF A =<3THENGOTO 70
 110 PLOT A,B," ":A=A-1:PLOT A,B,"h"
 115 IF SCRN(A-1,B)=108 THENGOTO40000
 120 IF SCRN(A,B-1)=105 THENMARK=MARK+10:PLOT A,B-1," "
 125 IF SCRN(A,B-1)=106 THENMARK=MARK+50:PLOT A,B-1," "
 130 IF SCRN(A,B-1)=107 THENMARK=MARK+100:PLOT A,B-1,"
"
 131 IF SCRN(A+1,B-1)=99THENPLOT A+1,B,"c"
 132 IF SCRN(A+1,B+1)=99THENPLOT A+1,B,"c"
 133 IF SCRN(A,B-1)=102THENGOTO 50000
 134 GOSUB 500:GOSUB600:GOSUB700:GOSUB800
 135 A$=KEY$
 140 IFA$<>"C"THENGOTO145
 141 IF A-4<=3THENGOTO145
 142 PLOT A,B," ":A=A-4:PLOTA,B,"h"
 145 IF A$="X"THENGOTO 200
 150 IF A$="/"THENGOTO300
 155 IF A$="'"THENGOTO400
 190 GOTO 100
```

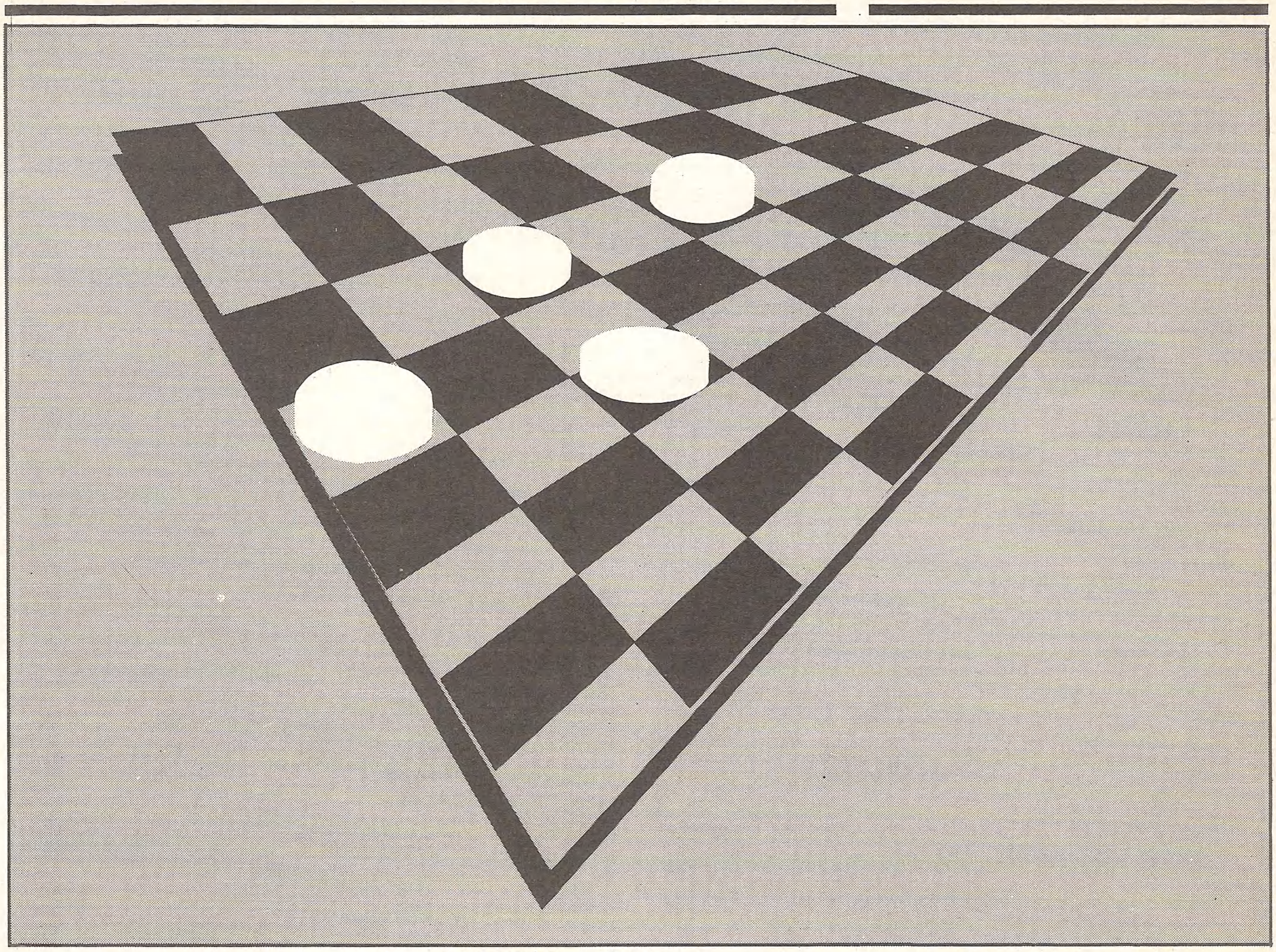
```
 200 REM *** ALIEN - RIGHT **
 205 IF A =>36THENGOTO70
 210 PLOT A,B," ":A=A+1:PLOT A,B,"h"
 215 IF SCRN(A+1,B)=108THENGOTO 40000
 220 IF SCRN(A,B-1)=105THENMARK=MARK+10:PLOT A,B-1," "
 225 IF SCRN(A,B-1)=106THENMARK=MARK+50:PLOT A,B-1," "
 230 IF SCRN(A,B-1)=107THENMARK=MARK+100:PLOT A,B-1," "
 231 IF SCRN(A-1,B-1)=99THENPLOT A-1,B,"c"
 232 IF SCRN(A-1,B+1)=99THENPLOT A-1,B,"c"
 233 IF SCRN(A,B-1)=102THENGOTO 50000
       235 GOSUB 500:GOSUB 600:GOSUB 700:GOSUB 800
       240 A$=KEY$
       245 IFA$<>"C"THENGOTO250
       246 IF A+4>=35THENGOTO250
       247 PLOTA,B," ":A=A+4:PLOTA,B,"h"
       250 IF A$="Z"THENGOTO100
       255 IF A$="/"THENGOTO300
       260 IF A$="'"THENGOTO400
 290 GOTO 200
 300 REM *** ALIEN - DOWN ***
 305 IF SCRN(A,B+1)<>99THENGOTO70
 310 PLOT A,B,"c":B=B+1:PLOT A,B,"h"
 315 GOSUB 500:GOSUB 600:GOSUB700:GOSUB 800
 390 GOTO 300
 400 REM *** ALIEN - UP ***
 405 FORQ=1TO4
 406 IF SCRN(A,B-1)<>99THENGOTO70
 410 PLOT A,B,"c":B=B-1:PLOT A,B,"h"
 415 IF SCRN(A,B-1)=104THENGOTO40000
 420 NEXTQ
 435 GOTO 70
 500 REM ** BONUS AND SHIP **
 505 PLOT C,D,"  ":PLOTE,F,"  ":C=C+1:E=E+1:PLOT C,D,Q$
:PLOT E,F,W$
 510 PLOT 2,1,4:PLOT 2,2,4:PLOT37,1,1:PLOT 37,2,1
 515 IF C>=35THENPLOT C,D,"  ":PLOT E,F,"  ":E=3:C=3
 520 PLOT 8,24," SCORE":PLOT 16,24,STR$(MARK):PLOT 21,2
4," LIVES"
 525 PLOT 29,24,STR$(LIVES)
 529 P=P-30
 530 PLOT 12,26,"--BONUS--":PLOT 23,26,LEFT$(STR$(P),5)
 535 IF P=0THENGOTO45000
 590 RETURN
 600 REM *** BOMB -1 ***
 605 PLOT ABS(G),H," "
 610 IF ABS(G)>=35ORABS(G)<=3THENG=-G:H=H+4
 615 G=G+1:PLOT ABS(G),H,"1"
 616 IF H>=20THENPLOT ABS(G),H," ":G=4:H=3
 620 IF SCRN(ABS(G-1),H-1)=99THENPLOT ABS(G-1),H,"c"
 625 IF SCRN(ABS(G-1),H+1)=99THENPLOT ABS(G-1),H,"c"
 630 IF SCRN(ABS(G+1),H)=104THENGOTO40000
 635 IF SCRN(ABS(G),H+1)=99THENO=INT(RND(1)*2):IFO=1THE
NGOTO 660
 650 RETURN
 660 FORQ=1TO4
 665 PLOT ABS(G),H,"c":H=H+1:PLOT ABS(G),H,"1"
 669 IF SCRN(ABS(G),H+1)=104 THENGOTO40000
 670 NEXTQ
 675 G=-G
 690 GOTO 600
 700 REM *** BOMB -2 ***
 705 PLOTABS(I),J," "
 710 IF ABS(I)>=35ORABS(I)<=3THENI=-I:J=J+4
 711 I=I+1:PLOT ABS(I),J,"1"
 715 IF SCRN(ABS(I-1),J+1)=99THENPLOT ABS(I-1),J,"c"
 716 IF J>=20THENPLOTABS(I),J," ":I=4:J=3
 720 IF SCRN(ABS(I-1),J-1)=99THENPLOT ABS(I-1),J,"c"
 725 IF SCRN(ABS(I+1),J)=104THENGOTO40000
 730 IF SCRN(ABS(I),J+1)=99THENO=INT(RND(1)*2):IF O=1TH
ENGOTO760
 750 RETURN
 760 FORQ=1TO4
 765 PLOT ABS(I),J,"c":J=J+1:PLOT ABS(I),J,"1"
 770 NEXTQ
 775 I=-I
 790 GOTO 700
 800 REM *** BOMB -3 ***
 805 PLOT ABS(K),L," "
 810 IF ABS(K)>=35OR ABS(K)<=3THENK=-K:L=L+4
 811 K=K+1:PLOT ABS(K),L,"1"
 815 IF SCRN(ABS(K-1),L+1)=99THENPLOT ABS(K-1),L,"c"
 816 IF L>=20THENPLOT ABS(K),L," ":K=4:L=3
 820 IF SCRN(ABS(K-1),L-1)=99THENPLOT ABS(K-1),L,"c"
 825 IF SCRN(ABS(K+1),L)=104THENGOTO40000
 830 IF SCRN(ABS(K),L+1)=99THENO=INT(RND(1)*2):IFO=1THE
NGOTO840
 ---.-----
  850 RETURN
  860 FORQ=1TO4
  865 PLOTABS(K),L,"c":L=L+1:PLOT ABS(K),L,"1"
  867 IF SCRN(ABS(K),L+1)=104THENGOTO40000
  870 NEXTQ
  875 K=-K
  1012 GOTO800
  10000 REM *** REDEFINITION ***
  10010 FOR A=0TO95:READD:POKE 46080+A+(8*ASC("a")),D:NE
 XTA
  10020 DATA 63,33,33,33,33,33,33,63
  10030 DATA 33,51,30,12,12,30,51,33
  10040 DATA 33,33,33,63,63,33,33,33
  10050 DATA 03,07,13,25,49,63,62,60
  10060 DATA 48,56,44,38,35,63,31,15
  10070 DATA 60,62,63,31,15,07,03,01
  10080 DATA 15,31,63,62,60,56,48,32
  10090 DATA 12,30,63,45,33,51,30,12
  10100 DATA 30,12,63,63,63,63,12,30
  10110 DATA 03,03,06,12,12,24,48,48
  10120 DATA 30,30,33,45,18,18,30,63
  10130 DATA 30,45,45,63,63,45,45,30
 19999 RETURN
 20000 REM *** INSTUCTIONS ***
 20010 CLS:PAPER6:INK4
 20020 PLOT 5,3," A L I E N":PLOT 5,4," A L I E N ":PLO
T 5,3,14:PLOT 5,4,14
 20030 PLOT 3,7," THE OBJECT OF THIS GAME IS TO HELP"
 20040 PLOT 3,9,"THEN LITTLE ALIEN GET BACK TO ITS"
 20050 PLOT 3,11,"SPACE SHIP (WHICH IS AT THE TOP"
 20060 PLOT 3,13," OF THE SCREEN). YOU DO THIS "
 20070 PLOT 3,15," BY CLIMBING LADDERS AND "
 20080 PLOT 3,17,"AVOIDING THE BOMBS THAT THE "
 20090 PLOT 3,19,"SPACE SHIP DROPS AT YOU"
 20100 PLOT 3,21,"THIS IS A SAFTY PRECAUTION ONLY"
 20110 PLOT 3,24,"   PRESS ANY KEY FOR CONTROLS"
 20120 PLOT 2,24,4:PLOT 0,24,16
 20130 GETA$
 20140 CLS:PAPER4:INK2
 20150 PLOT 19,6,"'":PLOT 18,4,"UP"
 20160 PLOT 9,13,"Z":PLOT 6,11,"RIGHT"
 20170 PLOT 29,13,"X":PLOT 27,11,"LEFT"
 20180 PLOT 19,19,"/":PLOT 17,21,"DOWN"
 20190 PLOT 5,23," PRESS ANY KEY TO START !"
 20200 PLOT 19,13,"C":PLOT 17,11," JUMP"
 20210 PLOT 1,6,1:PLOT 1,4,0:PLOT 1,13,1:PLOT 1,11,0:PL
OT1,19,1:PLOT 1,21,0
 20230 GETA$
 29999 RETURN
 30000 REM *** SET UP SCREEN ***
 30005 CLS:PAPER0:INK1
```

```
 30010 PLOT 1,0,"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
"
 30020 PLOT1,1,"a                                     a"
 30030 PLOT1,2,"a                                     a"
 30040 PLOT1,3,"a     c             c         c       a"
 30048 PLOT1,4,"a bbbbcbbbbbbbbbbbbbcbbbbbbbbbcbbbbbb a"

 30060 PLOT1,5,"a     c             c         c       a"

 30070 PLOT1,6,"a     c    i        c j       c  k    a"

 30080 PLOT1,7,"a     c     c       c    c    c       a"

 30090 PLOT1,8,"a bbbbbbbbbbcbbbbbbbbbbbbcbbbbbbbbbbb a"

 30100 PLOT1,9,"a           c            c            a"

 30110 PLOT1,10,"a    k      c     j      c       j    a
"
 30120 PLOT1,11,"a    c      c     c      c       c    a
"
 30130 PLOT1,12,"a bbbcbbbbbbbbbbbbcbbbbbbbbbbbbbbcbbb a
"
 30140 PLOT1,13,"a    c            c              c    a
"
 30150 PLOT1,14,"a    c     j      c  i           ci   a
"
 30160 PLOT1,15,"a    c         c  c   c          c    a
"
 30170 PLOT1,16,"a bbbbbbbbbbbbbcbbbbbbcbbbbbbbbbbbbbb a
"
 30180 PLOT1,17,"a              c      c               a
"
 30190 PLOT1,18,"a        j     c     jc           j   a
"
 30200 PLOT1,19,"a              c      c               a
"
 30210 PLOT1,20,"a bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb a
"
 30220 PLOT1,21,"a                                     a
"
 30230 PLOT1,22,"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
"
 30240 FORA=4TO20STEP4:PLOT 2,A,2:PLOT 37,A,1:NEXTA
 39999 RETURN
 40000 REM *** THE END? ***
 40010 LIVES=LIVES-1:IF LIVES<=0THENGOTO45000
 40020 CLS:PAPER2:INK4:PLOT 5,12," YOU LOSE A LIFE"
 40030 T$="12345678912345678912345123459876598765987654
3219876543211231199911991"
 40040 FORA=1TOLEN(T$)
 40050 W=VAL(MID$(T$,A,1))
 40060 PLAY 7,0,0,0:MUSIC 1,2,W,V :MUSIC 2,6,W,11:MUSIC
 2,3,W,5:NEXTA
 40070 PLAY 0,0,0,0:WAIT 100:GOTO50
 45000 REM *** THE END ***
 45005 CLS:PAPER5:INK6
 45010 PLOT 5,10,"THE END" :PLOT 5,9,"THE END":PLOT 4,9
,10:PLOT 4,10,10
 45020 T$="11991199191919876543223456789876543211234567
8999911119991919191919191"
 45030 FORA=1TOLEN(T$)
 45040 W=VAL(MID$(T$,A,1))
 45050 PLAY 7,0,0,0:MUSIC1,1,W,V :MUSIC 2,6,W,V :MUSIC
3,3,W,10
 45055 SOUND 6,W, V:SOUND 1,W,V :SOUND 3,W,10:SOUND 5,W
,8
 45060 NEXTA
 45070 PLAY 0,0,0,0:END
 50000 REM *** FINISH A SCREEN ***
 50005 PAPER4:INK0
 50010 CLS:P=P-1000:MARK=MARK+500
 50020 PLOT 5,10," GO ONTO ANOTHER SCREEN "
 50030 T$="12345676532234565425413245765421212456765421
234567654398765434598 76"
 50040 FOR A=1TOLEN(T$)
 50041 W=VAL(MID$(T$,A,1))
 50042 PLAY 7,0,0,100
 50045 MUSIC 1,3,W,V:MUSIC 2,1,W,V:MUSIC 3,6,W,V
 50060 NEXTA
 50070 PLAY 0,0,0,0
 50080 GOTO 50
 55000 REM *** VOLUME CONTROL ***
 55010 CLS:PAPER1:INK4
 55020 PLOT 5,5," VOLUME CONTROL":PLOT 5,6," VOLUME CON
TROL":PLOT5,5,14
 55030 PLOT 5,6,14
 55040 PLOT 2,8,"PICK THE VOLUME CONTROL "
 55050 PLOT 2,10,"BETWEEN THE NUMBERS (1 TO 15)":INPUT
A$
 55060 IF VAL(A$)<1OR VAL(A$)>15THENGOTO 55050
 55070 PLOT 10,20,"THE VOLUME IS ":PLOT 25,20,A$
 55080 PLOT 8,22," PRESS <H>TO CHANGE OR  <C> TO"
 55090 PLOT 12,24," CONTINUE"
 55100 GETA$
 55110 IF A$="H"THENGOTO55000
 59999 RETURN
```

Alan Northcott

REVERSI

Turn your life upside down with this game of strategy.



Reversi is a board game of tactical skills invented at the turn of the century. It was later 're-invented' and called Othello.

The game has been written in program form for one or two players. When input into your Oric a grid will appear onto which the players must place their pieces in an effort to 'capture' the ones of their opponent's. The strategy comes in when deciding where to place your piece. One of your counters either side of your opponent's (in any direction) allows you to convert your opponents piece to one of your own. The game ends when one player has all the pieces on the board. It is obviously harder to describe than to play but it is a fascinating and absorbing game.

Boolean Variables:

MOVING is TRUE when calling the taking routine to move a piece, FALSE when only requiring a value returned in

LIST OF VARIABLES USED:

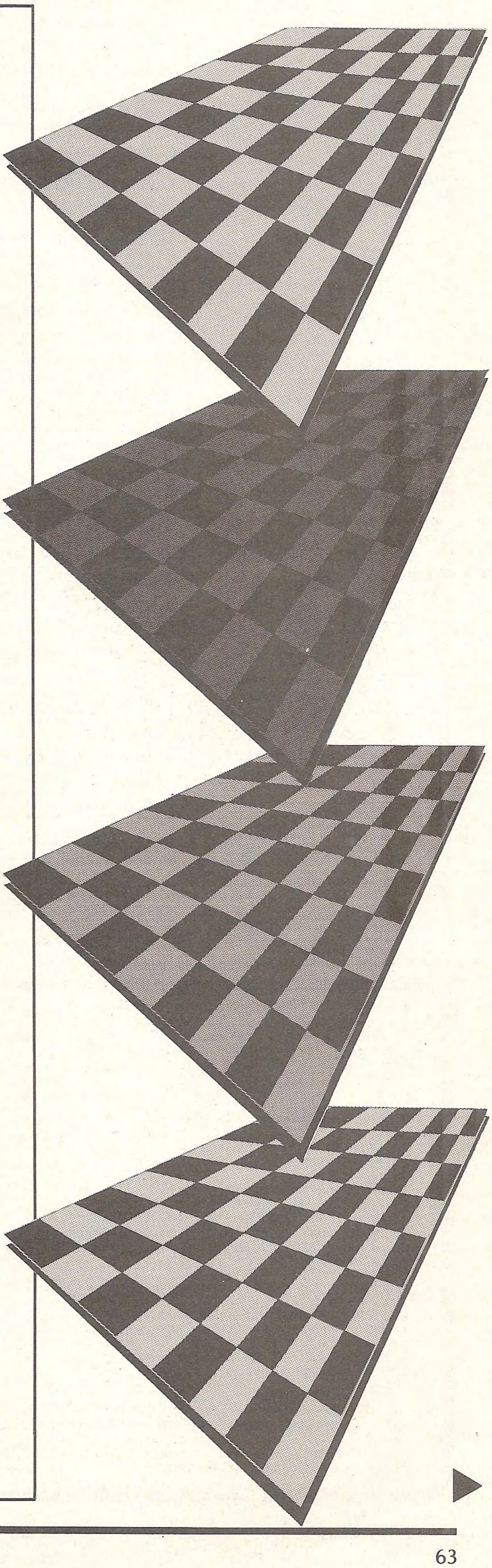
X,Y(Value 1-8)	Used generally in board setup, then to pass position to piece drawing routine
XC,YC(1-8)	Position of cursor used during player input
NX,NY(1-8)	New square under consideration in moving/taking routine
LX,LY,HX,HY(1-8)	Limits of initial consideration, to speed move
DX,DY(− 1.0 or 1)	Give direction from new square . . .
DIST	. . .and associated distance from new square
XBEST,YBEST(1-8)	'Best' player position found (so far)
PTS	Points 'scored' by that position
SCRE	Points 'scored' by currently considered position
WHITE,BLACK-(1or2)	1 for computer playing that colour. 2 for player
WP,BP	Number of white and black pieces
TAKE	Taking power of a move (no of pieces)

VIB	Period of SOUND commands
VAR	Change in pitch
T(X,Y)	Tactics array with a value for each square
BOARD(X,Y)	Board array, 1=Empty square, 2=White 3=Black
Also used:	N,TURN,ENEMY,A$,PL$,Z,COLOUR$

TAKE. GAMEOVER is TRUE if there are no more moves left, or the board is full, or end requested.
LEGAL is made FALSE if there is no legal move for a player
QUIET is TRUE if no sound is required.

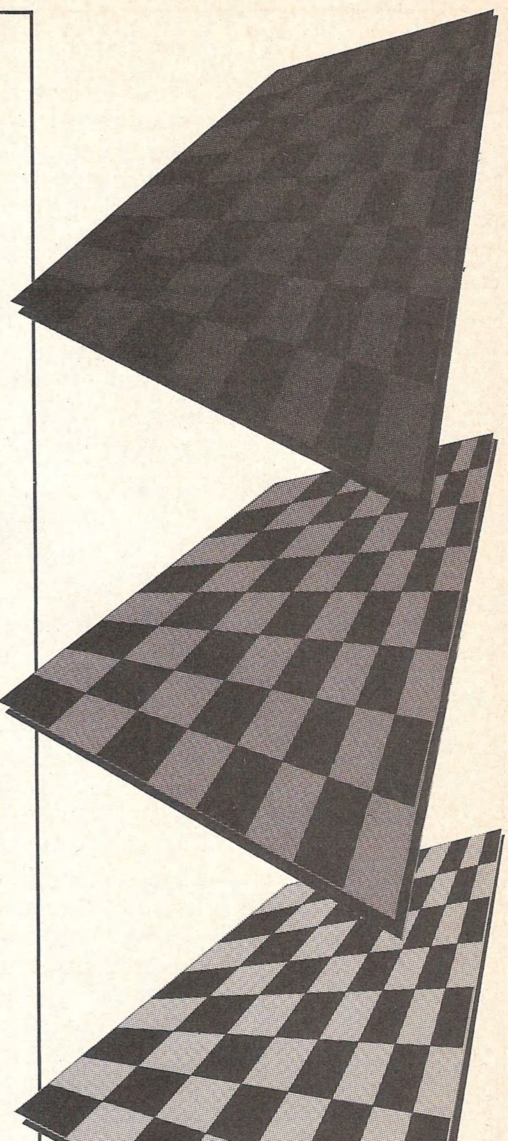
HOW IT RUNS

Lines 10-30	Set screen and HIMEM stops character corruption
Lines 40-50	Display title in double height, without cursor
Line 60	Goto routine to read tactical values
Lines 70-80	Goto instructions, if required
Lines 90-150	Ask and set if sound required
Line 160	Goto routine to set starting position
Line 170	Goto routine to determine number (type) of players
Line 180	Starts playing loop
Lines 190-225	Black's turn
Line 195	Displays score
Line 200	Checks for legal move and jumps if none
Lines 210-220	Make or get Black's move
Lines 230-260	White's turn — as Black's
Line 270	End of game
Lines 280-320	Another game or end
Line 1000	TAKE will be set to number of pieces captured by moving to NX, NY
Lines 1010-1020	For each direction in turn
Lines 1030-1050	. . . at increasing distance DIST
Line 1060	. . . until no longer enemy pieces
Line 1070	No enemy pieces encountered
Line 1080	Ensure next piece is own
Lines 1090-1210	Section to capture taken pieces
Lines 1120-1150	Set limits of move search
Lines 1220-1230	Next direction
Lines 2000-2460	Display two screens of instructions
Lines 3000-3130	Enter tactical values, read from the data statements, into the T(X,Y) array. The values, which are open to experimentation, range from 10 for the valuable corner, down to 0 & 1 for those squares adjacent to the corner, which may give it away.
Line 4000	Initialises HI RES display
Line 4010	Sets SOUND variables
Lines 4020-4090	Draw the board
Lines 4100-4140	Set all elements of the array BOARD (X,Y), which holds the current position, to 1 (empty).
Lines 4150-4190	Draw starting position
Line 4200	Number of pieces
Line 4210	Limits of move search
Lines 5000-5020	Determine number of players
Lines 5030-5040	Computer plays itself
Lines 5050-5060	Two players
Lines 5070-5110	One player select colour
Line 6000	Not moving yet
Line 6010	PTS (points) for the position set to an arbitrarily low figure
Lines 6020-6030	Considering each square in turn
Line 6040	If not empty, then skip
Lines 6050-6060	Determine taking power (if zero, skip)



▶

Line 6070	Move evaluation — the move is scored according to tactical value T(X,Y) a random element to give variation, the number of pieces taken by the move and the total played. TAKE *(XP+YP−32) is negative in the first half of the game, thus discouraging high taking moves early on, which are generally thought to be disadvantageous, and increasingly positive subsequently, to encourage higher takes towards the end. The evaluation is easily altered to change the importance of each element, and it is interesting to see the variation in style of play that can be achieved.
Lines 6080-6120	If best score so far, then the move is saved in XBEST, YBEST.
Lines 6130-6150	Call subroutine with MOVING=TRUE, to make the 'best' move
Lines 6155-6210	Option to interrupt game, as detailed in the instructions
Lines 7000-7060	Flash 'cursor' awaiting for a key press
Lines 7070-7110	Identify and act on cursor keys
Line 7120	RETURN pressed go to moving part of routine
Lines 7130-7160	Game options
Lines 7170-7230	check that the input move is valid then move it
Lines 8000-8100	Take each empty square in turn until a legal move is found, then make LEGAL=TRUE
Lines 9000-9100	Draw a white piece at X,Y
Lines 9010-9040	Sound
Lines 9050-9070	Draw 10 concentric circles
Lines 10000-10120	Draw a black piece at X,Y
Lines 10010-10040	Sound
Lines 10050-10070	Draw 9 concentric circles in background colour
Line 10080	Draws outer edge
Lines 11000-11040	Routine to display double edge height messages on bottom 3 lines in HI RES mode, with the previous message displayed (single height) above, as a reminder.



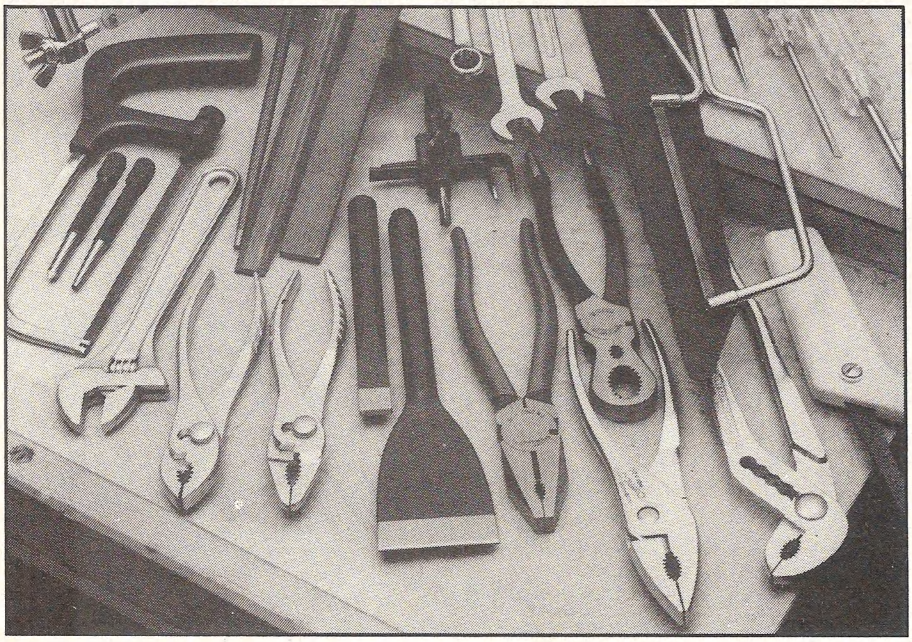
```
0 REM*******(*****(*******(**********
1 REM**                            **
2 REM**      ORIC REVERSI          **
3 REM**                            **
4 REM**      Alan Northcott        **
5 REM**                            **
6 REM**        May 1983            **
7 REM**                            **
8 REM********************************
10 CLS
20 TEXT
25 HIMEM#9800
30 PAPER1:INK7
40 PRINT:PRINT:PRINT:PRINT,,;
50 POKE#26A,90:PRINT"J  ORIC REVERSI"
60 PRINT:PRINT:PRINT:GOSUB3000'Tactics
70 PRINT,CHR$(27);"J Do you need instructions?"
80 A$=KEY$:IFA$="Y"THENGOSUB2000E~SEIFA$<>"N"THENGOTO80
90 CLS:PAPER0:INK2
100 PRINT:PRINT:PRINT
110 PRINT,,CHR$(27);"J Do you want sound?"
120 A$=KEY$
130 IFA$="Y"THEN QUIET=FALSE:GOTO160
140 IFA$="N"THEN QUIET=TRUE:GOTO160
150 GOTO120
160 GOSUB4000'Set up initial position
170 GOSUB5000'Number of players
180 REPEAT
190 : TURN=3:ENEMY=2
195 : M1$="SCORE: Black "+STR$(BP)+" - White "+STR$(WP):GOSUB11000
200 : GOSUB8000:IFLEGAL=FALSETHENM1$="Black has no move":GOSUB11000
:GOTO 225
210 : M1$="Black's move":GOSUB11000
220 : ON BLACK GOSUB 6000,7000
225 : IFGAMEOVERTHEN270
230 : TURN=2:ENEMY=3
235 : M1$="SCOPE: Black "+STR$(BP)+" - White "+STR$(WP):GOSUB11000
240 : GOSUB8000:IFLEGAL=FALSETHENM1$="White has no move":GOSUB11000
:GOTO270
250 : M1$="White's move":GOSUB11000
260 : ON WHITE GOSUB 6000,7000
270 UNTIL GAMEOVER
280 M1$="Another game?":GOSUB11000
290 A$=KEY$:IFA$=""THEN290
300 IFA$="Y"THEN160
310 IFA$<>"N"THEN280
320 TEXT:END

999 REM SUBROUTINE TO RETURN THE TAKING POWER OF A MOVE
1000 TAKE=0
1010 FOR DX=-1 TO 1
1020 :  FOR DY=-1 TO 1
1030 :    DIST=0
1040 :    REPEAT
1050 :      DIST=DIST+1
1060 :    UNTIL BOARD(NX+DX*DIST,NY+DY*DIST)<>ENEMY
1070 :    IF DIST=1 THEN 1220'NO TAKE IN THAT DIRECTION
1080 :    IF BOARD(NX+DX*DIST,NY+DY*DIST)<>TURN THEN 1220ELSETAKE=T
AKE+DIST-1
1090 : ~  IF NOT MOVING GOTO 1220
1100 :    IF TAKE<>DIST-1 THEN 1180
1108 :    X=NX:Y=NY
1120 :    IF X=LX THEN LX=X-1:IF LX=0 THEN LX=1
1128 :    IF X=HX THEN HX=X+1:IF HX=9 THEN HX=8
1140 :    IF Y=LY THEN LY=Y-1:IF LY=0 THEN LY=1
1150 :    IF Y=HY THEN HY=Y+1:IF HY=9 THEN HY=8
1160 :    ON TURN-1 GOSUB 9000,10000
1170 :    IF TURN=3THEN WP=WP+1 ELSE BP=BP+1
```

```
1180 :     FOR Z=1 TO DIST-1
1190 :       X=NX+DX*Z:Y=NY+DY*Z
1200 :       ON TURN-1 GOSUB 9000,10~00
1210 :     NEXT Z
1220 :   NEXT DY
1230 NEXT DX
1240 RETURN

1999 REM INSTRUCTIONS
2000 PRINTCHR$(4);CHR$(12)'Single height, clear screen
2010 PRINT
2020 PRINT" The game of Reversi, or Othello as"
2030 PRINT"it is now know~, is played on an 8x8"
2040 PRINT"board, play alternating until the"
2050 PRINT"board is full, or no move by either"
2060 PRINT"player is possible. If a player has"
2070 PRINT"no legal move, his turn is forfeit."
2080 PRINT" A move consists of 'placing' a piece"
2090 PRINT"on the board, next to an opponent's"
2100 PRINT"piece, so that the opponent's piece"
2110 PRINT"or pieces are trapped in a straight"
2120 PRINT"line between the newly placed piece"
2130 PRINT"and a player's piece already on the"
2140 PRINT"board. The piece(s) so trapped become"
2150 PRINT"the player's. Pieces may be captured"
2160 PRINT"both orthogonally and diagonally in"
2170 PRINT"several directions in the same move."
2180 PRINT" When the game is over, the player"
2190 PRINT"with the most pieces on the board is"
2200 PRINT"the winner."
2210 PRINT:PRINT"     PRESS ANY KEY TO CONTINUE"
2220 A$=KEY$:IFA$=""THEN2220
2230 PRINTCHR$(12):PRINT
2240 PRINT" To place a piece, move the flashing"
2250 PRINT"cursor to the required square using"
2260 PRINT"the arrow keys, then play by pressing"
2270 ~RINT"<RETURN> when the cursor is in the"
2280 PRINT"required position."
2290 PRINT:PRINT" You may choose 0,1 or 2 (human)"
2300 PRINT"players : -"
2310 PRINT"     0 - for demonstration, the"
2320 PRINT"  computer plays itself"
2330 PRINT"     1 - to play against the computer"
2340 PRINT"     2 - for a computer-moderated"
2350 PRINT"  game between two players, with"
2360 PRINT"  checks for legal moves and"
2370 PRI~T"  automatic scoring."
2380 PRINT:PRINT" Finally, pressing the following keys"
2390 PRINT"at any time will have the stated"
2400 PRINT"effect (after the move is complete):-"
2410 PRINT"  S - to stop temporarily, (press any"
2420 PRINT"        key to continue)"
2430 PRINT"  N - to start a new game"
2440 PRINT"  E - to end play."
2450 PRINT:PRINT"     PRESS ANY KEY TO START";
2460 A$=KEY$:IFA$=""THEN2460ELSEPRINTCHR$(4):RETURN

2999 REM ENTER TACTICAL VALUES
3000 FOR X=1TO4
3010 :  FORY=1TO4
3020 :    READ TACTIC
3030 :    T(X,Y)=TACTIC
3040 :    T(X,9-Y)=TACTIC
3050 :    T(9-X,9-Y)=TACTIC
3060 :    T(9-X,Y)=TACTIC
3070 :  NEXTY
3080 NEXTX
3090 DATA10,1,7,5
3100 DATA1,0,2,2
3110 DATA7,2,4,3
3120 DATA5,2,3,3
3130 RETURN

3999 REM SET UP BOARD AND DISPLAY
4000 HIRES:INK7:POKE#26A,10
4010 VIB=1500:VAR=1.06:M2$=""
4020 FOR X=25TO217 STEP24
4030 :  CURSETX,4,3
4040 :  DRAW0,191,1
4050 NEXTX
4060 FORY=4TO196STEP24
4070 :  CURSET25,Y,3
4080 :  DRAW191,0,1
4090 NEXTY
4100 FORX=1TO8
4110 :  FORY=1TO8
4120 :    BOARD(X,Y)=1' EMPTY SQUARE
4130 :  NEXTY
4140 NEXTX
4150 FORZ=4TO5
4160 :  X=Z
4170 :  Y=Z:GOSUB9000'DRAW WHITE
4180 :  Y=9-X:GOSUB10000'DRAW BLACK
4190 NEXTZ
4200 BP=2:WP=2
4210 LX=3:HX=6:LY=3:HY=6
4220 GAMEOVER=FALSE
4230 RETURN

4999 REM No & TYPE OF PLAYERS
5000 M1$="How many players?":GOSUB11000:GETPL$
5010 IF PL$<"0"ORPL$>"2"THENM1$="Enter 0,1 or 2":GOSUB11000:GOTO500
0
5015 M2$=M2$+"  "+PL$
5020 ONVAL(PL$)+1GOTO5030,5070,5050
5030 BLACK=1:WHITE=1'COMP.PLAYS BOTH
5040 RETURN
5050 BLACK=2:WHITE=2'TWO PLAYERS
5060 RETURN
5070 M1$="Will you play black or white?":GOSUB11000
5080 GET COLOUR$
5090 IF COLOUR$="B"THENBLACK=2:WHITE=1:M2$=M2$+" Black":RETURN
5100 IF ~OLOUR$="W"THENBLACK=1:WHITE=2:M2$=M2$+" White":RETURN
5110 GOTO5070

5999 REM COMPUTER'S MOVE
6000 MOVING=FALSE
6010 PTS=-1000
6020 FOR NX=LX TO HX
6030 :  FOR NY=LY TO HY
6040 :     IF BOARD(NX,NY)<>1 THEN 6110
6050 :     GOSUB 1000
6060 :     IF TAKE=0 THEN 6110
6070 :     SCRE=T(NX,NY)*20+RND(1)*10+TAKE*(XP+YP-32)'EVALUATE MOVE
6080 :     IF SCRE<PTS THEN 6110
6090 :     PTS=SCRE
6100 :     XBEST=NX:YBEST=NY
6108 :  NEXT NY
6120 NEXT NX
6130 NX=XBEST:NY=YBEST
6140 MOVING=TRUE
6150 GOSUB 1000
6155 IF BP+WP=64 THEN GAMEOVER=TRUE
6160 A$=KEY$:IFA$=""THENRETURN
6170 A=ASC(A$)
6180 IFA=83THENM1$="Press any key to re-start":GOSUB11000:GETA$:RET
URN
6190 IFA=78THEN GAMEOVER=TRUE:RETURN
6200 IF A=69 THEN TEXT:END
6210 RETURN

6999 REM ROUTINE TO INPUT PLAYER'S MOVE
7000 CURSET XC*24+10,YC*24-11,3
7010 CHAR126,0,1
7020 WAIT5
7030 CHAR126,0,0
7040 WAIT5,
7050 A$=KEY$:IF A$=""THEN7010
7060 IFBOARD(XC~YC)=2THEN ~HAR126,0,1
7070 A=ASC(A$)
7080 IF A=8 THEN XC=XC-1:IF XC<1THENXC=1
7090 IF A=9 THEN XC=XC+1:IF XC>8THENXC=8
7100 IF A=10THEN YC=YC+1:IF YC>8THENYC=8
7110 IF A=11THEN YC=YC-1:IF YC<1THENYC=1~
7120 IF A=13 AND BOARD(XC,YC)=1THEN 7170
7130 IFA=83THENM1$="Press any key to re-start":GOSUB11000:GET A$:GO
TO7010
7140 IF A=78 THEN GAMEOVER=TRUE:RETURN
7150 IF A=69 THEN TEXT:END
7160 GOTO7000
7170 NX=XC:NY=YC
7180 GOSUB 1000
7190 IF TAKE=0 THEN 7000
7200 MOVING=TRUE
7210 GOSUB 1000
7220 IF BP+WP=64 THEN GAMEOVER=TRUE
7230 RETURN
7999 REM ROUTINE TO CHECK FOR A LEGAL MOVE
8000 MOVING=FALSE:TAKE=0
8010 FOR NX=LX TO HX
8020 :  FOR NY=LY TO HY
8030 :     IF BOARD(NX,NY)<>1THEN8060
8040 :     GOSUB 1000
8050 :     IF TAKE<>0THEN XC=NX:YC=NY:NX=8:NY=8
8060 :  NEXT NY
8070 NEXT NX
8080 IF TAKE<>0 THEN LEGAL=TRUE:RETURN
8090 IF LEGAL=FALSE THEN GAMEOVER=TRUE ELSE LEGAL=FALSE
8100 RETURN

8999 REM DRAW A WHITE AT X,Y
9000 CURSETX*24+13,Y*24-8,3
9010 VIB=INT(VIB/VAR)
9020 IF QUIETTHEN9050
9030 SOUND1,VIB,0
9040 PLAY 1,0,1,1000
9050 FORN=1TO10
9060 :  CIRCLEN,1
9070 NEXTN
9080 BOARD(X,Y)=2
9090 WP=WP+1:BP=BP-1
9100 RETURN
9110 RETURN

9999 REM DRAW A BLACK AT X,Y
10000 CURSETX*24+13,Y*24-8,3
10010 VIB=INT(VIB*VAR)
10020 IFQUIETTHEN10050
10030 SOUND1,VIB,0
10040 PLAY 1,0,1,1000
10050 FORN=1TO9
10060 :  CIRCLEN,0
10070 NEXTN
10080 CIRCLE10,1
10090 BOARD(X,Y)=3
10100 BP=BP+1:WP=WP-1
10120 RETURN

10999 REM ROUTINE USED FOR BOTTOM 3 LI~ES - TO GIVE DOUBLE-HEIGHT M
ESSAGES
11000 PRINT:PRINT"  ";M2$
11010 PRINTCHR$(138)+" "+M1$
11020 PRINTCHR$(138)+" "+M1$;
11030 M2$=M1$
11040 RETURN
```

C Hamilton

TOOLKIT

Machine code causes untold problems for programmers, but here are some facilities to make writing and debugging it a little easier.



Here I have written about half a dozen facilities intended to make the task of debugging machine code much easier.

The program is driven from a menu and using it I hope you will all find a few problems you may have come across when machine code programming solved, as I did.

The first facility is a dissembler to convert the code in memory back to assembly language. A few points to note: immediate operands are displayed in hexadecimal, binary, and ASCII where appropriate. Relative branches are shown as destination addresses and signed decimal offsets. The display from the disassembler uses the following pattern:

Address: hex.codes mnemonic operand

The second facility is a memory dump which simply displays the contents of memory as follows:

Address: memory as hex. memory as ASCII

If the content of the memory location isn't a printable character, eg codes less than 32, they are represented in the ASCII section by a '.'.

The third facility is a search routine which will search through memory for a particular sequence of bytes. The sequence to be searched for can be entered either as a list of numbers, Hex or decimal, or as a string of characters. The routine is not particularly fast but it can search through at about 1K per minute, no matter how long the sequence is, up to a maximum of 255 bytes.

Next comes a facility for changing the contents of memory. An address and its contents are displayed and you have the following options: to change the contents, to set the address, to increment or decrement the address, to pad a block of memory with a value or, finally, to copy a block of memory from one location to another.

The fifth facility is simply a Hex to decimal (and vice versa) converter. The sixth however takes a two digit Hex number and returns the associated mnemonic, if there is one, or if you enter a mnemonic it will return the Hex codes associated with the addressing modes it uses.

On the disassembler, memory dump and sequence search routines there is an option to pause to output at any time by pressing SPACE. Pressing it again will restart, pressing ESC will abort the routine and pressing P will toggle the printer on and off. Having mentioned the printer I had better say that the output of the disassembler and memory dump can be directed to a printer and further, that the length of the printer line can be set from 10 to 255 characters.

Any time an address is requested it can be entered either as a decimal or hexadecimal number the Hex numbers preceded by '#'.

HOW IT RUNS

The running of the program together with a description of the important variables is explained in the REM statements so I feel it is unnecessary to duplicate them here.

HINTS ON CONVERSION

Any Oric specific POKEs or PEEKs have been explained in the REMs so that equivalents can be found. The only other Oric specific commands used are the MUSIC, PLAY and PING commands, these are used only to signal input errors and are not esential, byte error indications are more noticeable with them.

Obviously the disassembler and opcode/mnemonic converter work only for the 6502 but the other facilities apply equally well to all other processors.

Those machines which don't have REPEAT UNTIL can imitate it by changing the REPEAT and UNTIL lines as in the following example:

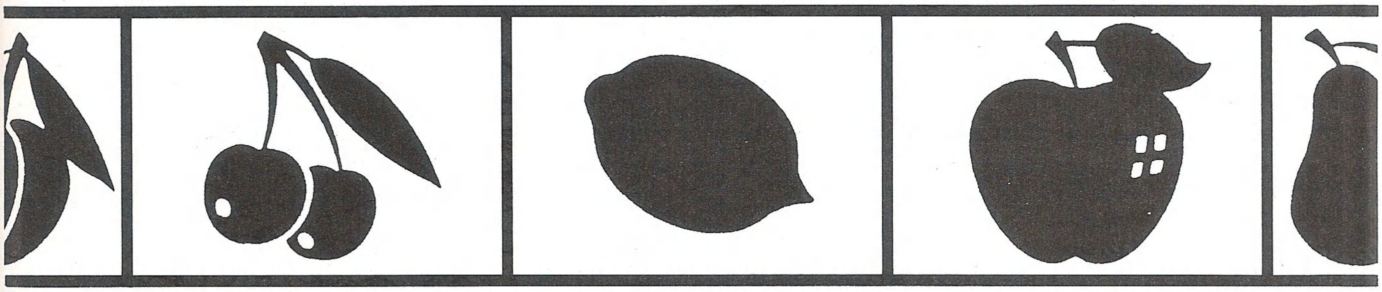
```
100 REPEAT        becomes   100 REM REPEAT
●                           ●
●                           ●
●                           ●
●                           ●
●                           ●
●                           ●
●                           ●
●                           ●
190 UNTIL A >10             190 IF A< = 10 THEN GOTO 100
```

```
1 REM ***********************
2 REM ** 6502 M/C Tool Kit **
3 REM **   © C Hamilton    **
4 REM **  Documented Copy  **
5 REM ***********************
6 REM
7 REM
9 REM ** Jump to Preliminaries **
10 GOSUB10000:GOTO9000
999 REM ** Disassembler **
1000 CLS:PRINT:PRINTTAB(27)"Disassemble":PRINT:PRINT
1009 REM ** AD = address , ST = start address
1010 AD=ST
1020 REPEAT
1029 REM ** D$ = data , BY = number of bytes in operand
1030 D$=D$(PEEK(AD)):BY=VAL(D$):TY=VAL(MID$(D$,5))
1038 REM ** TY = addressing mode , OU$ = output string
1039 REM ** Address
1040 OU$=RIGHT$("0000"+MID$(HEX$(AD),2),4)+":  "
1049 REM ** Hex
1050 FORI=0TOBY
1060 OU$=OU$+RIGHT$("00"+MID$(HEX$(PEEK(AD+I)),2),2)+" "
1070 NEXTI
1079 REM ** Pad with spaces
1080 FORI=LEN(OU$)TO16:OU$=OU$+" ":NEXTI
1089 REM ** Mnemonic
1090 AD=AD+1:OU$=OU$+"  "+MID$(D$,2,3)+" "
1099 REM ** Branch according to addressing mode
1100 IFTYTHENGOSUB1800+200*TY
1109 REM ** Move to next instruction
1110 AD=AD+BY
1119 REM ** Display string, to printer if set
1120 PRINTOU$:IFPRTHENLPRINTOU$:GOSUB9400
1129 REM ** Check for interrupt
1130 GOSUB9700
1140 UNTIL AD>FH
1149 REM ** Wait until ready
1150 GOSUB9900
1159 REM ** Return to main menu
1160 RETURN
1997 REM ** Addressing Modes and Operands**
1998 REM ** Immediate **
1999 REM ** Get operand
2000 IMM=PEEK(AD):X=IMM:Y%=128
2009 REM ** Hex
2010 OU$=OU$+"#"+RIGHT$("00"+MID$(HEX$(IMM),2),2)+"  "
2019 REM ** Binary
2020 FORI=1TO8
2030 IFY%>XTHENOU$=OU$+"0"ELSEOU$=OU$+"1":X=X-Y%
2040 Y%=Y%/2
2050 NEXTI
2059 REM ** ASCII, if appropriate
2060 IFIMM>31ANDIMM<127THENOU$=OU$+" '"+CHR$(IMM)+"'"
2070 RETURN
2199 REM ** Absolute **
2200 OU$=OU$+RIGHT$("0000"+MID$(HEX$(DEEK(AD)),2),4):RETURN
2399 REM ** Absolute,X **
2400 GOSUB2200:OU$=OU$+",X":RETURN
2599 REM ** Absolute,Y **
2600 GOSUB2200:OU$=OU$+",Y":RETURN
2798 REM ** Relative **
2799 REM ** OS = offset , check for negative
2800 OS=PEEK(AD)+1:IFOS>127THENOS=OS-256
2809 REM ** BR = destination address
2810 BR=AD+OS:OU$=OU$+RIGHT$("0000"+MID$(HEX$(BR),2),4)+"  ["
2819 REM ** Offset in signed decimal
2820 IFOS<0THENOU$=OU$+"-"ELSEOU$=OU$+"+"
2830 OU$=OU$+MID$(STR$(OS-1),2)+"]"
2840 RETURN
2999 REM ** Zero Page **
3000 OU$=OU$+RIGHT$("00"+MID$(HEX$(PEEK(AD)),2),2):RETURN
3129 REM ** 3 chars implies mnemonic
3199 REM ** Zero Page,X **
3200 GOSUB3000:OU$=OU$+",X":RETURN
3399 REM ** Zero Page,Y **
3400 GOSUB3000:OU$=OU$+",Y":RETURN
3599 REM ** Accumulator **
3600 OU$=OU$+"A":RETURN
3799 REM ** (Absolute Indexed) **
3800 OU$=OU$+"(":GOSUB2200:OU$=OU$+")":RETURN
3999 REM ** (Pre-indexed Indirect,X) **
4000 OU$=OU$+"(":GOSUB3000:OU$=OU$+",X)":RETURN
4199 REM ** (Post-indexed Indirect),Y **
4200 OU$=OU$+"(":GOSUB3000:OU$=OU$+"),Y":RETURN
4999 REM ** Opcode/Mnemonics **
5000 CLS:PRINT:PRINTTAB(25)"Opcode Mnemonics":PRINT:PRINT
5010 PRINT"  Enter either:":PRINT:PRINT
5020 PRINT"      a two digit hex number, Nb. '#'":PRINT
5030 PRINT"      not required or":PRINT:PRINT
5040 PRINT"      a three character mnemonic.":PRINT:PRINT
5050 PRINT"  Enter FF to finish.":PRINT:PRINT:PRINT
5100 REPEAT
5109 REM ** Get input, remove leading '#'s
5110 INPUT">:  ";A$:PRINT:IFASC(A$)=35THENA$=MID$(A$,2)
5119 REM ** Only allow 2 or 3 characters
5120 IFLEN(A$)<2ORLEN(A$)>3THENPRINT"Invalid.":PRINT:GOTO5110
5129 REM ** 3 characters implies mnemonic
5130 IFLEN(A$)=3THENGOSUB5200:GOTO5160
5138 REM ** Opcode given
5139 REM ** Check for legal value
5140 IFVAL("#"+A$)=0ANDA$<>"00"THENPRINT"Invalid.":PRINT:GOTO5110
5149 REM ** Reinstate leading '#'
5150 N$="#"+A$:GOSUB5300
5160 PRINT:PRINT
5170 UNTIL A$="FF"
5179 REM ** Return to main menu
5180 GOTO9005
5198 REM ** Mnemonic given
5199 REM ** Search for matches within opcode data
5200 FORN=0TO255
5209 REM ** AM$ = addressing mode , BY$ = number of bytes in operand
5210 GOSUB5500:IFOP$=A$THENPRINTH$"  "AM$"  "BY$
5220 NEXTN
5230 RETURN
5299 REM ** Opcode given
5300 N=VAL(N$):GOSUB5500
5309 REM ** Check for unassigned
5310 IFOP$="???"THENPRINT"Unassigned."ELSEPRINTOP$"  "AM$"  "BY$
5320 RETURN
5498 REM ** Convert opcode / mnemonic
5499 REM ** D$ = data , OP$ = mnemonic ,H$ = opcode
5500 D$=D$(N):OP$=MID$(D$,2,3):H$=RIGHT$("00"+MID$(HEX$(N),2),2)
5509 REM ** AM$ = addressing mode
5510 IFOP$="???"THENAM$=""ELSEAM$=AM$(VAL(MID$(D$,5)))
5519 REM ** BY$ = number of bytes in operand
5520 IFOP$="???"THENBY$=""ELSEBY$=CHR$(ASC(D$)+1)+" bytes"
5530 RETURN
5999 REM ** Memory Dump **
6000 CLS:PRINT:PRINTTAB(28)"Memory Dump":PRINT:PRINT
6009 REM ** UP = number of bytes per line, depends on line length
6010 IFPRTHENUP=8*INT((PL-8)/32)ELSEUP=8
6015 IFUP>32THENUP=32
6019 REM ** AD = start address, multiple of UP
6020 AD=INT(ST/UP)*UP
6030 REPEAT
6040 A$="":OU$=RIGHT$("0000"+MID$(HEX$(AD),2),4)+": "
6050 FORI=1TOUP
6059 REM ** D = code
6060 D=PEEK(AD):AD=AD+1
6069 REM ** Bytes in hex
6070 OU$=OU$+RIGHT$("00"+MID$(HEX$(D),2),2)+" "
6079 REM ** A$ = bytes in ASCII, where appropriate
6080 IFD>31ANDD<127THENA$=A$+CHR$(D)ELSEA$=A$+"."
6090 NEXTI
6099 REM ** Hex followed by ASCII
6100 OU$=OU$+" "+A$
6109 REM ** Display, to printer if set
6110 PRINTOU$:IFPRTHENLPRINTOU$:GOSUB9400
6119 REM ** Check for interrupt
6120 GOSUB9700
6130 UNTIL AD>FH
6139 REM ** Wait until ready
6140 GOSUB9900
6149 REM ** Return to main menu
6150 RETURN
6499 REM ** Poke Memory **
6500 CLS:PRINT:PRINTTAB(27)"Poke Memory":PRINT:PRINT
6510 PRINT"  An address is shown, enter:":PRINT:PRINT
6520 PRINT"     two digit hex number to be poked":PRINT
6530 PRINT"     S, set address":PRINT
6540 PRINT"     cursor left, decrement address":PRINT
6550 PRINT"     cursor right, increment address":PRINT
6555 PRINT"     P, pad memory with value":PRINT
6560 PRINT"     M, move block of memory":PRINT
6565 PRINT"     ESC, return to main menu":PRINT:PRINT:PRINT
6569 REM ** I0$,I1$ = strings of allowed input characters
6570 I0$="0123456789ABCDEFSPM"+CHR$(8)+CHR$(9)+CHR$(27)
6580 I1$=LEFT$(I0$,16)+CHR$(127)+CHR$(13)
6600 REPEAT
6609 REM ** Display address
6610 PRINTRIGHT$("0000"+MID$(HEX$(PA),2),4)" = ";
6614 REM ** Display current contents
6615 PRINTRIGHT$("00"+MID$(HEX$(PEEK(PA)),2),2)"  >:  ";
6619 REM ** Get command, check for ESC
6620 IN$=I0$:GOSUB6900:IFIN=22GOTO6750
6629 REM ** S, set address
6630 IFIN=17THENGOSUB6800:GOTO6740
6634 REM ** P, pad memory
6635 IFIN=18THENGOSUB6850:GOTO6740
6639 REM ** M, move block of memory
6640 IFIN=19THENGOSUB6870:GOTO6740
6644 REM ** cursor left/right decrement/increment address
6645 IFIN>19THENPA=PA-(IN=20)*(PA>0)+(IN=21)*(PA<#FFFF):PRINT:GOTO6740
6649 REM ** Add leading '#'
6650 N$="#"+A$:PRINTN$;:IN$=I1$
6659 REM ** Get another hex digit
6660 REPEAT
6669 REM ** Get character, check for RETURN
6670 GOSUB6900:IFIN=18GOTO6720
6679 REM ** Limit to '#' + two digits
6680 IFLEN(N$)=3ANDIN<17THENMUSIC1,1,1,0:PLAY1,0,1,800:GOTO6670
6689 REM ** Only delete if at least one digit there
6690 IFIN=17ANDLEN(N$)>1THENN$=LEFT$(N$,LEN(N$)-1):PRINTA$;
6699 REM ** Add on next digit
6700 IFIN<17THENN$=N$+A$:PRINTA$;
6719 REM ** Repeat until RETURN pressed
6720 UNTILA$=CHR$(13)
6730 PRINT:N=VAL(N$):POKEPA,N:PA=PA-(PA< FFFF)
6740 PRINT:PRINT
6749 REM ** Repeat until ESC pressed
6750 UNTILA$=CHR$(27)
6759 REM ** Return to main menu
6760 GOTO9005
6798 REM ** Change address
6799 REM ** Get input, force leading '#'
6800 INPUT"address = ";N$:IFASC(N$)<>35THENN$="#"+N$
6809 REM ** Set new address
6810 PA=VAL(N$)
6819 REM ** Only allow character if it is in IN$
6820 RETURN
6848 REM ** Pad memory
6849 REM ** Get start, finish addresses and value
6850 PRINT"PAD":GOSUB9800:INPUT"          value = ";N$
6854 REM ** Force leading '#' on value
6855 IFASC(N$)<>35THENN$="#"+N$
6859 REM ** Pad memory between limits
6860 N=VAL(N$):FORAD=STTOFH:POKEAD,N:NEXTAD
6865 RETURN
6868 REM ** Move block of memory
6869 REM ** Get addresses of block to be moved
6870 PRINT"MOVE BLOCK":PRINT:PRINT"From:":GOSUB9800:PRINT:PRINT"To:":PRINT
6874 REM ** Get destination addresses
6875 INPUT" Start address = ";DS$:DS=VAL(DS$)
6879 REM ** Check for valid address
6880 IFDS<0ORDS>#FFFFTHENMUSIC1,1,1,0:PLAY1,0,1,800:GOTO6875
6884 REM ** Transfere block byte by byte
6885 FORAD=STTOFH:POKEDS,PEEK(AD):DS=DS+1:NEXTAD
```

```
6890 RETURN
6899 REM ** Get character
6900 REPEAT
6910 GETA$:IN=0
6920 FORI=1TOLEN(IN$)
6930 IFA$=MID$(IN$,I,1)THENIN=I
6940 NEXTI
6950 UNTILIN>0
6960 RETURN
6999 REM ** Find Sequence **
7000 CLS:PRINT:PRINTTAB(27)"Find Sequence":PRINT:PRINT
7010 PRINT"  To find a sequence of bytes in memory":PRINT
7020 PRINT"type in the sequence as one of these :-":PRINT:PRINT
7030 PRINT"    a list of Decimal numbers,":PRINT
7040 PRINT"    a list of Hex. numbers, # not needed,"
7050 PRINT"    a String of characters, max. 255":PRINT:PRINT
7060 PRINT"Which option D,H or S ? ";
7069 REM ** Get sequence format option
7070 REPEAT:GETA$:UNTILA$="D"ORA$="H"ORA$="S"
7079 REM ** S$ = source sequence
7080 PRINTA$:S$="":IFA$="S"THENGOSUB8500ELSEGOSUB8000
7089 REM ** Get start, finish addresses
7090 GOSUB9800:AD=ST
7099 REM ** Just the first occurence or all of them ?
7100 PRINT:PRINT:PRINT"First or all occurences (F/A) ? ";
7110 REPEAT:GETA$:UNTILA$="F"ORA$="A"
7119 REM ** FO = flag, 0 = all, 1 = first only
7120 IFA$="F"THENFO=1ELSEFO=0
7130 CLS:PRINT:PRINT"Searching . . . . . . .":PRINT:PRINT
7139 REM ** TB$ = cursor control string
7140 TB$=CHR$(11):FORI=1TO16:TB$=TB$+CHR$(9):NEXTI
7498 REM ** Build string from memory
7499 REM ** AF = address of next byte after current string
7500 F$="":AF=AD+LEN(S$)-1
7510 FORI=ADTOAF:F$=F$+CHR$(PEEK(I)):NEXTI
7514 REM ** Switch off cursor
7515 POKE#26A,PEEK(#26A)AND254
7519 REM ** For each match
7520 REPEAT
7529 REM ** Step through memory
7530 REPEAT
7539 REM ** FF = found flag
7540 PRINT"Match found at":FF=FALSE
7550 REPEAT
7559 REM ** Display address of start of current string
7560 PRINTTB$RIGHT$("0000"+MID$(HEX$(AD),2),4)
7569 REM ** If matched then set flag
7570 IFF$=S$THENFF=TRUE
7579 REM ** Increment addresses and update string
7580 AD=AD+1:AF=AF+1:F$=MID$(F$,2)+CHR$(PEEK(AF))
7589 REM ** Check for interrupt
7590 GOSUB9700
7600 UNTIL FF OR AD>FH OR AF=#FFFF
7606 REM ** Stop searching if:
7607 REM ** 1: only looking for first and found it
7608 REM ** 2: past given finish address
7609 REM ** 3: run out of memory to search !
7610 UNTIL (FFANDFO) OR AD>FH OR AF=#FFFF
7614 REM ** Switch cursor back on
7615 POKE#26A,PEEK(#26A)OR1
7619 REM ** Blank out last display if not a match
7620 IFNOTFFTHENPRINTCHR$(11);"
7629 REM ** Wait until ready
7630 GOSUB9900
7639 REM ** Return to main menu
7640 GOTO9005
7999 REM ** Get sequence as a list of numbers
8000 CLS:PRINT:PRINT"Type in sequence, ** to finish, press":PRINT
8010 PRINT"RETURN after each:":PRINT:PRINT
8020 REPEAT
8029 REM ** Get number, check for last
8030 INPUT">:  ";N$:IFN$="**"GOTO8080
8039 REM ** If hex chosen as default then force leading '#'
8040 IFA$="H"ANDASC(N$)<>35THENN$="#"+N$
8050 N=VAL(N$)
8059 REM ** Check for legal value
8060 IFN<0ORN>255THENMUSIC1,1,1,0:PLAY1,0,1,800:GOTO8030
8069 REM ** Add onto source string
8070 S$=S$+CHR$(N)
8079 REM ** Stop when '**' entered
8080 UNTIL N$="**"
8090 RETURN
8499 REM ** Get sequence as a string
8500 CLS:PRINT:PRINT"Type in string (Max. length = 255)":PRINT:PRINT
8510 REPEAT
8519 REM ** Get character, check for RETURN
8520 GETA$:IFA$=CHR$(13)GOTO8570
8524 REM ** Check for CTRL/T and toggle CAPS
8525 IFA$=CHR$(20)THENPRINTA$;:GOTO8520
8529 REM ** Refuse to accept more than 255 characters
8530 IFLEN(S$)=255ANDA$<>CHR$(13)THENPING:GOTO8520
8534 REM ** If sting is empty and DEL pressed then ignore
8535 IFLEN(S$)=0ANDA$=CHR$(127)GOTO8520
8539 REM ** Remove last character from string if DEL pressed
8540 IFA$=CHR$(127)ANDLEN(S$)>1THENS$=LEFT$(S$,LEN(S$)-1):GOTO8560
8545 IFA$=CHR$(127)ANDLEN(S$)=1THENS$="":GOTO8560
8548 REM ** Add character onto source string
8549 REM ** If character is a control then display space instead
8550 S$=S$+A$:IFASC(A$)<32THENA$=" "
8560 PRINTA$;
8569 REM ** Stop when RETURN pressed
8570 UNTIL A$=CHR$(13)
8580 RETURN
8999 REM ** Main Menu **
9000 REPEAT
9005 CLS:PRINT:PRINT
9010 PRINT:PRINT"The following options are available:":PRINT:PRINT
9020 PRINTSPC(10)"1  Disassemble":PRINT:PRINTSPC(10)"2  Dump memory":PRINT
9030 PRINTSPC(10)"3  Find sequence":PRINT:PRINTSPC(10)"4  Poke memory":PRINT
9040 PRINTSPC(10)"5  Convert hex/dec":PRINT:PRINTSPC(10)"6  Opcode/mnemonic":
PRI NT
9045 PRINTSPC(10)"7  Select printer":PRINT:PRINTSPC(10)"8  Goodbye":PRINT
9050 PRINT:PRINT"Which do you want ? ";
9059 REM ** Get option, BOOP if illegal
9060 REPEAT:GETO$:O=VAL(O$):IFO<1ORO>8THENMUSIC1,1,1,0:PLAY1,0,1,800
9065 UNTILO>0ANDO<9:PRINTO$
9069 REM ** Branch to choice
9070 ONOGOTO9100,9200,7000,6500,9500,5000,9300
9079 REM ** Check for Goodbye
9080 UNTIL O=8
9084 REM ** Clear top line
9085 FORI=#BB80TO#BBA0:POKEI,32:NEXTI
9089 REM ** Reset keyboard/VDU vector and END
9090 CLS:POKE#26A,3:END
9099 REM ** Get start, finish addresses and branch to disassembler
9100 GOSUB9800:GOSUB1000:GOTO9005
9199 REM ** Get start, finish addresses and branch to memory dump
9200 GOSUB9800:GOSUB6000:GOTO9005
9299 REM ** Set up printer
9300 CLS:PRINT:PRINT"Press:":PRINT:PRINT"       P to toggle printer,":PRINT
9302 PRINT"       L to change line length,":PRINT:PRINT"       RETURN to return
to menu":PRINT
9304 PRINT:PRINT"  Nb. pressing P resets top of page.":PRINT:PRINT
9306 PRINT"Which do you want ? ";
9309 REM ** Get command
9310 REPEAT:GETA$:UNTILA$="P"ORA$="L"ORA$=CHR$(13):PRINTA$:PRINT:PRINT
9319 REM ** Return to main menu if RETURN pressed
9320 IFA$=CHR$(13)GOTO9005
9329 REM ** Toggle printer on/off if P pressed and reset line count
9330 IFA$="P"THENPR=ABS(PR-1):LL=1
9339 REM ** Get new line length if L pressed
9340 IFA$="L"THENREPEAT:INPUT"Length = ";PL$:PL=VAL(PL$):UNTILPL>10ANDPL<256
9344 REM ** Display printer status
9345 GOSUB9350:GOTO9300
9349 REM ** Set new LPRINT line length
9350 POKE#31,PL:D$="Printer ":IFPRTHEND$=D$+"ON"ELSED$=D$+"OFF"
9360 D$=D$+"/"+MID$(STR$(PL),2)+"    "
9369 REM ** Display printer status on top line
9370 FORI=1TOLEN(D$):POKE#BB81+I,ASC(MID$(D$,I,1)):NEXTI
9380 RETURN
9399 REM ** Printer paging routine
9400 LL=LL+1
9409 REM ** Move paper over perforations
9410 IFLL>64THENLPRINT:LPRINT:LL=1
9420 RETURN
9499 REM ** Dec/Hex Converter **
9500 CLS:PRINT:PRINTTAB(24)"Dec/Hex Converter":PRINT:PRINT
9510 PRINT"  To use simply type in the number to be"
9520 PRINT"converted, all hexadecimal numbers must":PRINT
9530 PRINT"begin with a '#'.":PRINT
9540 PRINT"  Entering 0 will cause a return to the":PRINT
9550 PRINT"main menu.":PRINT:PRINT:PRINT
9559 REM ** TB$ = cursor control string
9560 TB$=CHR$(11):FORI=1TO15:TB$=TB$+CHR$(9):NEXTI
9600 REPEAT
9609 REM ** Get number
9610 INPUT":  ";N$:N=VAL(N$):PRINTTB$;
9614 REM ** Check for valid input
9615 IFN=0ANDLEFT$(N$,2)<>"#0"ANDASC(N$)<>48THENPRINT"Invalid.":PRINT:GOTO9610
9620 IFN>#FFFFTHENPRINT"Too large.":PRINT:GOTO9610
9625 IFN<-32767THENPRINT"Too small.":PRINT:GOTO9610
9629 REM ** If negative decimal then complement with #FFFF
9630 IFN<0THENN=#FFFF+N+1
9634 REM ** If hex then display +ve and -ve decimal equivalents
9635 IFASC(N$)=35ANDN>#80ANDN<#100THENPRINTN;"or ";-(#100-N);:GOTO9660
9640 IFASC(N$)=35THENPRINTN;:IFN>#8000THENPRINT"or ";-(#FFFF-N+1);
9649 REM ** If decimal then display hex equivalent
9650 IFASC(N$)<>35THENPRINT"#"RIGHT$("0000"+MID$(HEX$(N),2),4);
9660 PRINT:PRINT
9669 REM ** Stop when 0 is input
9670 UNTIL N=0
9679 REM ** Return to main menu
9680 GOTO9005
9699 REM ** Check for interrupt
9700 K$=KEY$:IFK$=""THENRETURN
9709 REM ** If ESC pressed then abort
9710 IFK$=CHR$(27)THENAD=FH+1:RETURN
9719 REM ** If P pressed then toggle printer and reset line count
9720 IFK$="P"THENPR=ABS(PR-1):LL=1:GOSUB9350:RETURN
9729 REM ** Wait until SPACE or ESC pressed
9730 REPEAT:GETK$:UNTILK$=" "ORK$=CHR$(27)
9739 REM ** If ESC pressed then abort
9740 IFK$=CHR$(27)THENAD=FH+1
9750 RETURN
9799 REM ** Get start and finish address
9800 PRINT:PRINT
9809 REM ** Get start address
9810 INPUT" Start address = ";ST$:ST=VAL(ST$)
9814 REM ** Check if valid, 0 <= start <= #FFFF (65535)
9815 IFST<0ORST>#FFFFTHENMUSIC1,1,1,0:PLAY1,0,1,800:PRINT"Invalid.":GOTO9810
9819 REM ** Get finish address
9820 INPUT"Finish address = ";FH$:FH=VAL(FH$)
9824 REM ** Check if valid, start <= finish <= #FFFF (65535)
9825 IFST>FHORFH>#FFFFTHENMUSIC1,1,1,0:PLAY1,0,1,800:PRINT"Invalid.":GOTO9820
9830 RETURN
9899 REM ** WaiLEFT$ until SPACE pressed
9900 PRINT:PRINT:PRINTTAB(21)"Press SPACE to continue.";
9910 REPEAT:GETA$:UNTILA$=" "
9920 RETURN
9996 REM ** Preliminaries
9997 REM ** Preliminaries
9998 REM ** Location #26A = keyboard/VDU vector
9999 REM ** Set 40 column white/black
10000 CLS:POKE#26A,39:PAPER0:INK7
10004 REM ** Clear top line and make it white/red
10005 FORI=#BB80TO#BBA0:POKEI,32:NEXTI:DOKE#BB80,#711
10008 REM ** PR = printer on/off flag , PL = printer line length
10009 REM ** Location #268 holds Y co-ord of cursor, set to line 9
10010 PR=0:PL=80:GOSUB9350:POKE#268,9
10020 PRINT:PRINTTAB(25)"6502 M/C TOOL KIT":PRINTTAB(25)"-----------------":PRINT
10030 PRINTTAB(24)CHR$(96)" C Hamilton 1:6:83":PRINTTAB(24)"-------------------"
10034 REM ** Switch off cursor
10035 PRINTCHR$(17)
10040 DIM D$(255),AM$(12)
10049 REM ** D$() = opcode data
10050 FORI=0TO255:READ D$(I):NEXTI
10059 REM ** AM$() = addressing modes
10060 FORI=0TO12:READ AM$(I):NEXTI
10070 WAIT200
10074 REM ** Switch cursor back on
10075 PRINTCHR$(17)
10080 RETURN
10995 REM ** Opcode data
10996 REM ** Chars  Meaning
10997 REM **     1   number of bytes in operand
10998 REM **   2-4   mnemonic
10999 REM **   5-6   addressing mode
11000 DATA 0BRK0 ,1ORA11,0???  ,0???  ,0???  ,1ORA6 ,1ASL6 ,0???
11010 DATA 0PHP0 ,1ORA1 ,0ASL9 ,0???  ,0???  ,2ORA2 ,2ASL2 ,0???
11020 DATA 1BPL5 ,1ORA12,0???  ,0???  ,0???  ,1ORA7 ,1ASL7 ,0???
11030 DATA 0CLC0 ,2ORA4 ,0???  ,0???  ,0???  ,2ORA3 ,2ASL3 ,0???
11040 DATA 2JSR2 ,1AND11,0???  ,0???  ,1BIT6 ,1AND6 ,1ROL6 ,0???
11050 DATA 0PLP0 ,1AND1 ,0ROL9 ,0???  ,2BIT2 ,2AND2 ,2ROL2 ,0???
11060 DATA 1BMI5 ,1AND12,0???  ,0???  ,0???  ,1AND7 ,1ROL7 ,0???
11070 DATA 0SEC0 ,2AND4 ,0???  ,0???  ,0???  ,2AND3 ,2ROL3 ,0???
11080 DATA 0RTI0 ,1EOR11,0???  ,0???  ,0???  ,1EOR6 ,1LSR6 ,0???
11090 DATA 0PHA0 ,1EOR1 ,0LSR9 ,0???  ,2JMP2 ,2EOR2 ,2LSR2 ,0???
11100 DATA 1BVC5 ,1EOR12,0???  ,0???  ,0???  ,1EOR7 ,1LSR7 ,0???
11110 DATA 0CLI0 ,2EOR4 ,0???  ,0???  ,0???  ,2EOR3 ,2LSR3 ,0???
11120 DATA 0RTS0 ,1ADC11,0???  ,0???  ,0???  ,1ADC6 ,1ROR6 ,0???
11130 DATA 0PLA0 ,1ADC1 ,0ROR9 ,0???  ,2JMP10,2ADC2 ,2ROR2 ,0???
11140 DATA 1BVS5 ,1ADC12,0???  ,0???  ,0???  ,1ADC7 ,1ROR7 ,0???
11150 DATA 0SEI0 ,2ADC4 ,0???  ,0???  ,0???  ,2ADC3 ,2ROR3 ,0???
11160 DATA 0???  ,1STA11,0???  ,0???  ,1STY6 ,1STA6 ,1STX6 ,0???
11170 DATA 0DEY0 ,0???  ,0TXA0 ,0???  ,2STY2 ,2STA2 ,2STX2 ,0???
11180 DATA 1BCC5 ,1STA12,0???  ,0???  ,1STY7 ,1STA7 ,1STX8 ,0???
11190 DATA 0TYA0 ,2STA4 ,0TXS0 ,0???  ,0???  ,2STA3 ,0???  ,0???
11200 DATA 1LDY1 ,1LDA11,1LDX1 ,0???  ,1LDY6 ,1LDA6 ,1LDX6 ,0???
11210 DATA 0TAY0 ,1LDA1 ,0TAX0 ,0???  ,2LDY2 ,2LDA2 ,2LDX2 ,0???
11220 DATA 1BCS5 ,1LDA12,0???  ,0???  ,1LDY7 ,1LDA7 ,1LDX8 ,0???
11230 DATA 0CLV0 ,2LDA4 ,0TSX0 ,0???  ,2LDY3 ,2LDA3 ,2LDX4 ,0???
11240 DATA 1CPY1 ,1CPM11,0???  ,0???  ,1CPY6 ,1CPM6 ,1DEC6 ,0???
11250 DATA 0INY0 ,1CPM1 ,0DEX0 ,0???  ,2CPY2 ,2CPM2 ,2DEC2 ,0???
11260 DATA 1BNE5 ,1CPM12,0???  ,0???  ,0???  ,1CPM7 ,1DEC7 ,0???
11270 DATA 0CLD0 ,1CPM4 ,0???  ,0???  ,0???  ,2CPM3 ,2DEC3 ,0???
11280 DATA 1CPX1 ,1SBC11,0???  ,0???  ,1CPX6 ,1SBC6 ,1INC6 ,0???
11290 DATA 0INX0 ,1SBC1 ,0NOP0 ,0???  ,2CPX2 ,2SBC2 ,2INC2 ,0???
11300 DATA 1BEQ5 ,1SBC12,0???  ,0???  ,0???  ,1SBC7 ,1INC7 ,0???
11310 DATA 0SED0 ,2SBC4 ,0???  ,0???  ,0???  ,2SBC3 ,2INC3 ,0???
11319 REM ** Addressing modes
11320 DATA Implied,Immediate,Absolute,"Absolute,X","Absolute,Y"
11330 DATA Relative,Zero page,"Zero page,X","Zero page,Y"
11340 DATA Accumulator,(Absolute indexed)
11350 DATA "(Pre-indexed indirect,X)","(Post-indexed indirect),Y"
```

FRUIT MACHINE

Here is a chance to play one of those addictive games in a situation where it won't cost you all your small change.



How many times have you been tempted into an arcade to play the fruit machines?? I for one won't deny this. The only problem with these gambling beasts is that I always end up losing most if not all, of my precious money.

The aim of the game, if you're not familiar with it, is to achieve combinations of certain symbols. Obviously some symbols are harder to come by than others. As an example, with my game the chances of getting a cherry is one in about three and four cherries are worth £1, and the chances of getting a bar is one in eleven and four bars is the jackpot of £20. So you see it is quite hard to get four bars in one spin. The most I've ever got is four apples worth £2!!!

You might think that it is difficult to get any combination of any symbol but you are helped by the 'HOLD' feature which enables you to freeze the reels for the next spin.

When it it time to spin the reels the message 'PRESS ANY KEY' will appear on the top of the screen. Any key can be pressed (except the two SHIFT keys) and after pressing the key, four symbols will appear in the four boxes, accompanied by four notes.

When the message 'HOLD' comes below the reels you can freeze one, two, three, or all four reels for the next spin simply by pressing the numbers '1', '2', '3', or '4' corresponding to which reel you want held. Having chosen which reels you want held, press the 'RETURN' key to spin the reels.

Another feature of the program is the GAMBLE facility

RUNdown

Lines	Action
3	Toggles cursor and keyclick off.
7-40	Defines A$ as message to be printed at top of screen and POKEs it there.
55	Asks if instructions wanted and jumps to instructions if wanted.
60-80	Defines main variables for money, dimensions arrays for contents of reels and HOLD.
108	Chooses random number for each reel.
109-118	Converts values in the array 'A' so that they are low enough for printout.
144-160	Checks if there has been a win.
170	Checks if win is not equal to 0 if it is, carry on. If not GOTO win routine.
171	Checks to see if money is 0. If so GOTO 'out of money' routine.
188	Random number for hold. Can be reduced to enable more frequent holds.
200	Determines value of win depending on the contents of the array 'A' (i.e. the value of the reels).
1000-1999	Printout of the fruit and sound effects.
2000-2040	Reading and poking of user defined graphics.
3000-3120	Hold routine.
9000-9092	DATA for graphics, including fruits and screen printout graphics.
9100-9220	Printout of screen, including reels, money, and border.
9300-9900	Instructions.
10000-10085	Sound effects for win. Printout of money won and gamble/collect option.
10086-10091	Adds win to money, turns off sound and jumps back to main program.
10100-10350	Gambling routine with sound effects.
60000-62010	Out of money routine and option to play again.

which enables you to increase your win, anything from 20p to £1, or £2 to £20. The system for gambling is that when you win, pressing the 'G' key starts the gamble routine. You will then see one of the two columns of numbers start flashing different numbers. Pressing the 'SPACE' bar will stop these numbers and the final number will be shown. If no number is shown they you have lost the gamble and no money has been won. If the number shown is higher than the original win, for example if you won 20p and decide to gamble and after pressing the space bar 50p is displayed you can either collect this win by pressing the 'C' key, which will then be added on to your money, or gamble again. If you decide to gamble again the same thing will happen at last time and after pressing the space bar if, say, 20p comes up this will be added to your money as if you collected.

The 'C' key enables you to cancel the hold. That is to say if you have made a mistake in the HOLD feature you can start the hold again.

There is no GAMBLE facility on 10p wins, £10 or £20 wins although it is possible to gamble up to £10 and £20.

If you win £1, collect it because of the way the gamble is done it is not possible to win over this amount with the gamble if you win less than £1.

It is not worth holding one of anything because there is not much chance of three more coming up.

VARIABLES USED

A$	Holds title for first part of program.
MO	Total money left.
A	Dimensioned array for value of each reel.
ZZ	Holds value for each reel held.
F.G.H.	Counter & control variables.
WIN	Holds value of money won each spin.
X	Value of first reel to work out win.
A$,K$	Used in main program and subroutines to 'GET' the keyboard.

CONVERSION CLUES

X,Y,N where X is the column number, Y is the row number and N is a number less than 8. The colour codes are as follows:

0 — Black
1 — Red
2 — Green
3 — Yellow
4 — Blue
5 — Magenta
6 — Cyan
7 — White

or PRINT CHR$ (N) where N is a number less than eight which also appears to the above.

The Oric screen is 40 columns by 27 rows. PLOT X,Y,Z$ PRINTS the string Z at position X,Y.

Address 48000 is the beginning of the Oric screen.

Most of the other commands are fairly standard BASIC and most, if not all of the REMs can be ommitted up into the various sections.

Please note that all characters are designed on a 6×8 basis as the Oric characters are 6×8.

All other control codes can be ommitted as they will only control double height, flashing etc.

Please note: if you have a 16K ORIC-1 you will be able to run the program only if the command 'GRAB' is given directly before typing in and before loading.

```
1 CLS :PAPER0:INK7
3 PRINTCHR$(6);CHR$(17)
5 CLS
7 A$="  **WRITTEN BY J.B.HADLEY JANUARY 1984**"
20 FORF=48000TO48039
30 POKEF,ASC(MID$(A$,F-47999,1))
35 NEXT
40 POKE48001,5
50 POKE48041,5:POKE48040,10
55 PLOT0,5,CHR$(12)+CHR$(4)+"DO YOU WANT INSTRUCTIONS?":GETA$:IFA$=
"Y"THEN9300
56 PLOT0,5,"                                      "
60 MO=190
70 DIMA(4)
80 DIMXX(4)
89 REM ******************************
90 REM ******** MAIN-PROGRAM ********      **********************
********
100 GOSUB9000
108 W=FRE(""):FORF=1TO4:A(F)=INT(RND(1)*22):A(F)=INT(A(F)/2):XX(F)=
0:NEXT F
109 FORF=1TO4
110 IFXX(F)<>0THEN118
111 IFA(F)=1ORA(F)=0ORA(F)=2THENA(F)=1
112 IFA(F)=3ORA(F)=4THENA(F)=2
113 IFA(F)=5ORA(F)=6THENA(F)=3
114 IFA(F)=7ORA(F)=8THENA(F)=4
115 IFA(F)=9THENA(F)=5
116 IFA(F)=10THENA(F)=6
118 NEXTF
120 GOSUB1000
130 WIN=0
144 IFA(1)=1THENWIN=10
145 IFA(1)=1ANDA(2)=1THENWIN=20
150 IFA(1)=A(2)ANDA(2)=A(3)THENGOSUB200
160 IFA(1)=A(2)ANDA(2)=A(3)ANDA(3)=A(4)THENGOSUB260
170 IFWIN<>0THENGOSUB10000
171 IFMO=0THEN60000
175 PLOT5,3,CHR$(6)+CHR$(12)+"P R E S S  A N Y  K E Y"+CHR$(7)+CHR$
(10)
176 PLOT20,19,CHR$(10)+"MONEY -"+STR$(MO)+" P"+CHR$(7)
177 PLOT20,20,CHR$(10)+"MONEY -"+STR$(MO)+" P"+CHR$(7)
180 GETA$
181 MO=MO-10
183 AA=FRE("")
184 PLOT20,20,CHR$(10)+"MONEY -"+STR$(MO)+" P"+CHR$(7)
185 PLOT20,19,CHR$(10)+"MONEY -"+STR$(MO)+" P"+CHR$(7)
187 PLOT5,3,"                              "
188 IFINT(RND(1)*5)=3THENGOSUB3000
190 GOTO108
200 X=A(1)
210 IFX=1ORX=2THENWIN=50
220 IFX=3ORX=4THENWIN=100
230 IFX=5THENWIN=200
240 IFX=6THENWIN=500
250 RETURN
260 X=A(1)
270 IFX=1ORX=2THENWIN=100
280 IFX=3ORX=4THENWIN=200
290 IFX=5THENWIN=1000
300 IFX=6THENWIN=2000
310 RETURN
998 REM *************************
999 REM ******* PRINTOUT ********         **********************
****
1000 FORH=1TO4
1005 F=A(H)
1010 G=(H*8)-2
1020 IFF=1THENPLOTG,7,CHR$(2)+"ab"+CHR$(7):PLOTG,8,CHR$(1)+"cd"+CHR
$(7)
1030 IFF=2THENPLOTG,7,CHR$(3)+"ij"+CHR$(7):PLOTG,8,CHR$(3)+"kl"+CHR
$(7)
1040 IFF=3THENPLOTG,7,CHR$(5)+"mn"+CHR$(7):PLOTG,8,CHR$(5)+"op"+CHR
$(7)
1050 IFF=4THENPLOTG,7,CHR$(2)+"ef"+CHR$(7):PLOTG,8,CHR$(1)+"gh"+CHR
$(7)
1060 IFF=5THENPLOTG,7,CHR$(2)+"qr"+CHR$(7):PLOTG,8,CHR$(2)+"st"+CHR
$(7)
1070 IFF=6THENPLOTG,7,CHR$(12)+"uv"+CHR$(8):PLOTG,8,CHR$(12)+"wx"+C
HR$(8)
1080 SOUND1,1000-(H*200),0
1082 SOUND2,0984-(H*200),0
1084 SOUND3,0972-(H*200),0
1085 PLAY7,0,1,500
1090 WAIT10
1100 NEXT
1105 WAIT50
1110 SOUND1,0,15
1999 RETURN
2000 RESTORE:FORF=46360 TO 46391
2010 READD:POKEF,D:NEXT
2020 FORF=46856TO47047:READD:POKEF,D:NEXT
```

```
2030 RETURN
2040 REM *****************************
2999 REM ******** H O L D ***********          *********************
********
3000 FORF=1TO4:XX(F)=0:NEXT
3007 PLOT5,11,CHR$(12)+"HOLD    HOLD    HOLD    HOLD"+CHR$(8)
3010 GETA$
3015 IFASC(A$)=13THEN3100
3016 IFA$="C"THENGOTO3000
3020 IFASC(A$)>52ORASC(A$)<49THEN3010
3040 IFA$="1"THENXX(1)=A(1):PLOT5,11,8:PLOT10,11,12
3050 IFA$="2"THENXX(2)=A(2):PLOT12,11,8:PLOT18,11,12
3060 IFA$="3"THENXX(3)=A(3):PLOT19,11,8:PLOT26,11,12
3070 IFA$="4"THENXX(4)=A(4):PLOT28,11,8
3080 GOTO3010
3100 FORF=1TO4
3110 IFXX(F)=0THENA(F)=INT(RND(1)*22):A(F)=INT(A(F)/2)
3111 NEXT
3115 PLOT5,11,"                                   "
3120 GOTO109
8998 REM ****************************
8999 REM ***** SCREEN PRINTOUT *****          *********************
*******
9000 DATA0,0,0,0,0,0,0,255,255,0,0,0,0,0,0,0,32,32,32,32,32,32,32,3
2
9001 GOSUB2000
9010 DATA1,1,1,1,1,1,1,1
9030 DATA0,0,0,0,1,2,4,8,3,4,28,36,4,4,4,4 ' CHERRY CHR
9040 DATA12,30,63,63,63,63,30,12,14,31,63,63,63,63,30,12  ' CHERRY
CHR2
9045 DATA0,0,28,62,31,15,15,7,4,8,18,22,47,14,28,48  ' APPLE CHR

~~9050 DATA7,15,31,63,63,63,31,15,56,60,62,63,63,63,62,60  ' APPLE C
HR2~~
9060 DATA 0,0,0,0,3,15,31,63,0,0,0,0,48,60,62,63     ' LEMON CHR
9070 DATA63,31,15,3,0,0,0,0,63,62,60,48,0,0,0,0     ' LEMON CHR2
9072 DATA1,3,7,15,25,27,63,62,32,48,56,60,38,46,63,31    ' ORANGE CH
R
9074 DATA62,63,27,24,15,7,3,1,31,63,54,6,60,56,48,32    ' ORANGE CHR
2
9080 DATA1,1,3,3,7,7,15,15,32,32,48,48,56,56,60,60     '  BELL CHR
9082 DATA31,31,63,63,31,1,3,3,62,62,63,63,62,32,48,48 ' BELL CHR2
9090 DATA3,18,59,18,3,0,51,42,0,36,14,36,0,0,44,42     ' BAR CHR
9092 DATA51,42,50,0,19,58,19,2,44,42,42,0,4,46,4,32    ' BAR CHR2
9100 PLOT1,1,"######################################":FORF=2TO26:PL
OT1,F,"%"
9110 PLOT38,F,"&":NEXT:PLOT1,26,"$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$"
9120 PLOT5,5,"######  ######  ######  ######":FORF=6TO9:PLOT4,F,2:P
LOT36,F,7
9130 PLOT5,F,"%    &  %    &  %    &  %    &":NEXT
9140 PLOT5,10,"$$$$$$  $$$$$$  $$$$$$  $$$$$$"
9160 PLOT4,15,CHR$(4)+" 20      1 "+CHR$(2)+"  ######### #########"+
CHR$(7)
9165 PLOT18,16,CHR$(2)+"%GAMBLE& %COLLECT&"
9170 PLOT4,17,CHR$(5)+"_10      50 P"+CHR$(7)+" $$$$$$$$$ $$$$$$$$$"+
CHR$(7)
9180 PLOT4,19,CHR$(6)+"_5       20 P"+CHR$(7)
9190 PLOT4,21,CHR$(3)+"_2       10 P"+CHR$(7)
9200 PLOT20,19,CHR$(10)+"MONEY -"+STR$(MO)+" P"+CHR$(7)
9210 PLOT20,20,CHR$(10)+"MONEY -"+STR$(MO)+" P"+CHR$(7)
9217 PLOT0,0,5
9220 RETURN
9298 REM ***************************
9299 REM ****** INSTRUCTIONS ******          *********************
******
9300 CLS:PRINT
9310 PRINT"  FRUIT MACHINE BY J.B.HADLEY 1984"
9315 GOSUB2000
9320 PRINT"  FRUIT MACHINE BY J.B.HADLEY 1984"
9330 PLOT1,2,10:PLOT1,1,10
9340 PRINT:PRINT"WELCOME TO 'FRUIT MACHINE'
9350 PRINT:PRINT"WRITTEN BY J.B.HADLEY IN JANUARY 1984"
9360 PRINT:PRINT"THE OBJECT IS TO SPIN THE 'REELS' TO  TRY TO OBTAI
N THE FOLLOWING ";
9370 PRINT:PRINT"COMBINATIONS"
9380 PRINT:PRINT:PRINT"PRESS ANY KEY FOR COMBINATIONS"
9390 GETA$
9400 CLS:PRINT
9410 PRINT"  C O M B I N A T I O N      W I N"
9415 PRINT"  $$$$$$$$$$$$$$$$$$$$$      $$$$$"
9420 PRINT:PRINT"  ab    ab    ##    ##       20P"
9430 PRINT"  cd    cd    $$    $$"
9440 PRINT:PRINT"  ab    ab    ab    ##       50P"
9450 PRINT"  cd    dc    dc    $$"
9460 PRINT:PRINT"  ef    ef    ef    ##       50P"
9470 PRINT"  gh    gh    gh    $$"
9480 PRINT:PRINT"  ij    ij    ij    ##       100P"
9490 PRINT"  kl    kl    kl    $$"
9500 PRINT:PRINT"  mn    mn    mn    ##       100P"
9510 PRINT"  op    op    op    $$"
9520 PRINT:PRINT"  qr    qr    qr    ##       200P"
9530 PRINT"  st    st    st    $$"
9540 PRINT:PRINT"  uv    uv    uv    ##       500P"
9550 PRINT"  wx    wx    wx    $$"
9560 PRINT:PRINT"PRESS ANY KEY TO CONTINUE"
9570 GETA$
9580 CLS
9590 PRINT:PRINT"  C O M B I N A T I O N      W I N"
9600 PRINT"  $$$$$$$$$$$$$$$$$$$$$      $$$$$".
9605 PRINT:PRINT"  ab    ##    ##    ##       10P "
9607 PRINT"  cd    $$    $$    $$"
9610 PRINT:PRINT"  ab    ab    ab    ab       100P"
9620 PRINT"  cd    dc    dc    cd"
9630 PRINT:PRINT"  ef    ef    ef    ef       100P"
9640 PRINT"  gh    gh    gh    gh"
9650 PRINT:PRINT"  ij    ij    ij    ij       200P"
9660 PRINT"  kl    kl    kl    kl"
9670 PRINT:PRINT"  mn    mn    mn    mn       200P"
9680 PRINT"  op    op    op    op"
9690 PRINT:PRINT"  qr    qr    qr    qr      1000P"
9700 PRINT"  st    st    st    st"
9710 PRINT:PRINT"  uv    uv    uv    uv      2000P"
9720 PRINT"  wx    wx    wx    wx"
9730 PRINT:PRINT"PRESS ANY KEY TO CONTINUE"
9740 GETA$:CLS
9750 PRINT:PRINT:PRINT"A HOLD FACILITY IS ALSO PROVIDED WHICHENABLE
S YOU TO FREEZE THE ";
9760 PRINT"REELS."
9770 PRINT:PRINT"YOU CAN DO THIS WHEN THE SYMBOL 'HOLD'APPEARS.YOU
HOLD THE REELS BY ";
9780 PRINT"PRESSING'1','2','3' OR '4' ACCORDING TO WHICH REEL YOU W
ANT HOLDING."
9790 PRINT:PRINT"WHEN YOU HAVE SELECTED THE REELS TO BEHELD PRESS T
HE 'RETURN KEY'."
9795 PRINT:PRINT:PRINT"IF YOU HAVE MADE THE WRONG HOLD THEN  YOU CA
N CANCEL THE HOLD BY "
9796 PRINT"PRESSING THE 'C' KEY."
9800 PRINT:PRINT:PRINT:PRINT"PRESS ANY KEY TO CONTINUE":GETA$:CLS
9810 PRINT:PRINT"WHEN YOU HAVE ONE OF THE COMBINATIONS SHOWN EARLIE
R YOU WILL WIN."
9820 PRINT:PRINT"YOU CAN EITHER COLLECT BY PRESSING 'C'OR GAMBLE BY
 PRESSING 'G'."
9830 PRINT:PRINT"IF YOU DECIDE TO COLLECT THE WIN WILL BE ADDED ONT
O YOUR MONEY."
9840 PRINT:PRINT"IF YOU GAMBLE ,PRESS 'SPACE' TO STOP  THE GAMBLE.T
HEN IT WILL SHOW THE
9850 PRINT"VALUE OF YOUR GAMBLE-IF YOU WIN THE   GAMBLE YOU CAN GAM
BLE AGAIN"
9860 PRINT:PRINT"IF YOU LOSE THE GAMBLE YOU WILL JUST  CARRY ON AS
IF NO WIN"
9870 PRINT:PRINT"IF YOU COME UP EVENS YOU WILL NOT BE  ABLE TO GAMB
LE AGAIN BUT THIS"
9880 PRINT"WILL BE ADDED ON TO YOUR MONEY...."
9890 PRINT:PRINT"YOU START WITH 200P":PRINT:PRINT"PRESS ANY KEY TO
START GAME":GETA$:CLS
9900 GOTO60
9998 REM ************************
9999 REM ******** W I N *********          *********************
****
10000 PLOT5,3,CHR$(4)+CHR$(12)+"WE HAVE A  W I N N E R!"+CHR$(7)+CH
R$(10)
10001 RET=0
10005 FORG=1TO3
10010 FORF=600TO0STEP-15:SOUND1,F,15:NEXT
10020 NEXT
10021 IFWIN=20THENPLOT11,19,CHR$(12)+" 20 P"+CHR$(8)
10022 IFWIN=50THENPLOT11,17,CHR$(12)+" 50 P"+CHR$(8)
10023 IFWIN=100THENPLOT11,15,CHR$(12)+" _1  "+CHR$(8)
10024 IFWIN=2000THENPLOT4,15,CHR$(12)+"_20"+CHR$(8)
10025 IFWIN=1000THENPLOT4,17,CHR$(12)+"_10"+CHR$(8)
10026 IFWIN=500THENPLOT4,19,CHR$(12)+"_5 "+CHR$(8)
10027 IFWIN=200THENPLOT4,21,CHR$(12)+"_2 "+CHR$(8)
10028 IFWIN=10THENPLOT12,21,CHR$(12)+"10 P"+CHR$(8)
10029 IFWIN>500ORWIN=10THEN10086
10030 PLOT18,16,CHR$(2)+"%       & %COLLECT&"
10040 FORF=1TO40STEP5:K$=KEY$:IFK$="G"THEN10100
10045 SOUND1,100+F*2,15
10050 IFK$="C"THEN10086
10055 NEXT
10070 PLOT18,16,CHR$(2)+"%GAMBLE& %       &"
10075 FORF=1TO40STEP5:K$=KEY$:IFK$="G"THEN10100
10078 SOUND1,100-F*2,15
10080 IFKEY$="C"THEN10086
10085 NEXT:GOTO10030
10086 SOUND1,0,15
10089 WAIT100:GOSUB9160
10090 IFRET=1THEN10110
10091 MO=MO+WIN:RETURN
10100 RET=1:GOTO10086
10105 K$=KEY$
10110 IFWIN<101THENG=12:H=5
10111 IFWIN>100THENG=4:H=8
10115 FORF=15TO21STEP2:PLOTG,F,0:PLOTG+H,F,7:NEXT
10118 F=23:GOTO10140
10120 F=INT(RND(1)*5)*2:F=F+15
10140 PLOTG,F,7:K$=KEY$:IFK$=" "THENGOTO10200
10145 SOUND1,INT(RND(1)*500),15
10150 PLOTG,F,0:PLOT25,23,7
10160 GOTO10120
10200 IFG=4THEN10300
10210 IFF=23THENW=0
10220 IFF=15THENW=100
10230 IFF=17THENW=50
10240 IFF=19THENW=20
10250 IFF=21THENW=10
10260 IFW<=WIN THEN RET=0:WIN=W:GOTO10086
10270 IFW=0THENWIN=0:GOTO10260
10280 WIN=W:GOTO10000
10300 IFF=23THENW=0
10310 IFF=15THENW=2000
10320 IFF=17THENW=1000
10330 IFF=19THENW=500
10340 IFF=21THENW=200
10350 GOTO10260
59998 REM **************************
59999 REM ****** OUT OF MONEY ******          ********************
*******
60000 FORF=1TO500STEP5
60010 SOUND1,F,0
60020 SOUND2,500-F,0
60040 NEXT
60050 SOUND1,0,0
60060 PRINTCHR$(6);CHR$(17)
60070 CLS:A$="SORRY YOU'VE RUN OUT OF CASH!!!!"
60080 GOSUB62000
60090 PRINT:PRINT:A$="DO YOU WANT TO TRY AGAIN?"
60100 GOSUB62000
60110 GETA$:IFA$="N"THEN61000
60120 PRINT:PRINT:A$="O.K. READY TO START...."
60130 GOSUB62000
60140 WAIT500
60150 RUN
61000 PRINT:PRINT:A$="O.K.THANKS FOR PLAYING 'FRUIT MACHINE'"
61010 GOSUB62000
61020 WAIT4E4:GOTO61020
62000 FORF=1TOLEN(A$):PRINTMID$(A$,F,1);:SOUND1,INT(RND(1)*500),15:
NEXTF
62010 SOUND1,0,0:RETURN
63998 REM ************************
63999 REM ***** T H E  E N D *****          *******************
*****
```

David Nowotnik

SANTA'S PROBLEM

A snowstorm has blown away all the labels on Father Christmas' parcels. We explain how you can help him.

In this program for the Oric 1, Father Christmas has a problem in delivering his presents.

Santa loads up his sleigh with 10 presents. They are all carefully labelled with the recipient's name, and these are shown on the screen, with the contents of each parcel, at the start of the game.

You get 30 seconds to memorise the list (you'll need to remember this to help Santa later), then off you go on his sleigh.

The journey south is depicted on the screen. The sleigh and reindeer will appear at the top of the screen, and you must imagine that you are above the sleigh, looking down at it.

The journey starts quietly enough, then all of a sudden you run into a snow storm with giant flakes of snow.

You have to help Santa to guide his sleigh through the snow storm, avoiding the snow flakes. You use the left and right cursor control key to move the sleigh.

Every time you hit a snow flake you'll hear a crash, and a point will be deducted from your eventual score. The sleigh won't be damaged by the snow, but the object is to minimise the number of collisions.

Once you are through snow storm, Father Christmas has to deliver the presents. But this is where he runs into another problem.

On the rough journey south, all the labels have been blown away, and he doesn't know which present to deliver to which house. You'll have to help him.

A list of all the presents appears on the screen. As you come to a house, Santa will tell you who lives there, you have to tell him which present to deliver

MAIN VARIABLES

A	Number of names and presents.
E0, D0	Indicator arrays.
K	A random number.
NL$	Array holding 10 selected names.
NM$	Array holding all available names.
PL	Indicator array.
PL$	Array holding the 10 selected presents.
PR$	Array holding all available presents.
SC	Score.
I	Counter.
I0	Sleigh character codes.
W	Input variable.
X,Y	Column position of sleigh.

by entering the number beside the present which you think should be delivered.

Once all 10 have been delivered, then you are shown your list, and the correct list. You'll get two points for every present delivered, and a point deducted for every snow flake you hit on the journey south.

The object of course, is to achieve a maximum score of 20. The game is a test of memory with the snow storm acting as a distraction.

In the program listing, there are 16 names and presents in data lines 5000 and above. In every game, 10 of these are selected at random.

To increase the interest in the game, you should add more names and presents.

To do this, add more DATA lines, each containing a name and present, and replace the number in the DATA line 5000 with the new total of names and presents.

HOW IT WORKS

10-110	Initialise main variables for guessing game.
120-170	PRINT 10 selected names and presents on screen.
190	30 second delay.
1000-1070	Initialise main variables for snow storm game.
1120-1130	Check for valid keypress.
1140-1160	Remove sleigh from screen.
1170	Scroll screen.
1180-1200	Check if hit by snowflake, and reprint sleigh.
1220-250	Print snowflakes.
1270-1350	Print score of snowflake game.
1440-1460	Start guessing game — initialise.
1470-1660	Print name, present list, enter, check selection.
1670-1780	Print player's selection, and correct list — award points.

```
10 REM   Christmas Snow Storm
20 REM     by David Nowotnik
30 REM        December, 1983
40 REM
50 REM  Initialise
60 REM
70 READA
80 DIMNM$(A),PR$(A)
90 DIMPL(A),NL$(10),PL$(10)
100 POKE618,10:PAPER7:INK0
110 FORI=1TOA:READNM$(I),PR$(I):NEXT
120 CLS:PRINT:PRINT,"Santa's Present List":PRINT:PRINT
50 FORI=1TO10
140 K=INT(RND(1)*A)+1:IFPL(K)=1THEN140
150 PRINTNM$(K),PR$(K)
160 NL$(I)=NM$(K):PL$(I)=PR$(K)
170 PL(K)=1:NEXTI
180 PRINT:PRINT:PRINT,"You have 30 seconds to",,,"remem
ber this lis
t"
190 PING:WAIT3000
200 PING:CLS
210 PRINT:PRINT:PRINT:PRINT"  O.K. Off we go with those
 presents"
220 WAIT500
1000 REM    Snow storm
1010 X=15:SC=0:T=0
1020 INK7:PAPER0:CLS
1050 FORI=1TO28:PRINT:NEXT
1060 DIMT(3)
1070 T(1)=160:T(2)=35:T(3)=35
1100 REPEAT
1110 Y=X
1120 IFPEEK(520)=172ANDX>0THENX=X-1
1130 IFPEEK(520)=188ANDX<38THENX=X+1
1140 FORJ=1TO3:REM Remove Sleigh from screen
1150 POKE48082+Y+40*J,32
1160 NEXT
1170 PRINT:T=T+1:REM Move up snowflakes
1180 FORJ=1TO3:REM Detect hit and reprint sleigh
1190 IFPEEK(48082+X+40*J)<>32THENSC=SC+1:EXPLODE
1200 POKE(48082+X+40*J),T(J):NEXTJ
1210 IFT<10ORT>90THENWAIT5:GOTO1260
1220 REM  Print snowflakes
1230 FORI=1TO2+INT(4*RND(1))
1240 POKE49042+INT(RND(1)*37),42
1250 NEXTI
1260 UNTILT=115
1270 PRINTCHR$(30):PRINT:PRINT:PRINT:PRINT:PRINT
1280 PRINT,"You are through the storm"
1290 PRINT:PRINT
1300 IFSC>1THEN1340
1310 IFSC=1THEN1330
1320 PRINT,"You avoided all snowflakes":GOTO1350
1330 PRINT,"You struck just 1 snowflake":GOTO1350
1340 PRINT,"You struck ";SC;" snowflakes"
1350 WAIT500 -
1360 PRINT:PRINT:PRINT" Now lets deliver the presents."
1370 PRINT:PRINT" Remember, Santa has forgotten which"
1380 PRINT"present belongs to which child. When"
1390 PRINT"you come to a house, Santa will give"
1400 PRINT"you the name, you have to select the"
1410 PRINT"right present."
1420 PRINT:PRINT:PRINT"    Press any key to start"
1430 IFPEEK(520)=56THEN1430
1440 PAPER7:INK0:CLS:PRINTCHR$(17)
1450 DIME(10),D(10)
1460~SC=-SC:T=0
1470 FORI=1TO10
1480 CLS
1490 K=1+INT(RND(1)*10):IFE(K)=1THEN1490
1500 PRINT:PRINT" Child's name is ";NL$(K)
1510 E(K)=1
1520 PRINT:PRINT:PRINT
1530 FORJ=1TO10
1540 IFD(J)<>0THEN1560
1550 PRINTJ,PL$(J)
1560 NEXT
1570 PRINT:PRINT:PRINT"    Enter the number of the toy
for"
1580 PRINTNL$(K)
1590 PRINT:PRINT
1600 INPUTW
1610 IFW<1ORW>10ORW<>INT(W)THEN1600
1620 IFD(W)<>0THEN1600
1630 PRINT:PRINT"Present ";PL$;" is delivered"
1640 IFW=KTHENT=T+1
1650 WAIT400:D(W)=K
1660 NEXTI
1670 CLS:PRINT:PRINT"  Compare your list on the right w
ith"
1680 PRINT"the correct list in the centre."
1690 PRINT:PRINT
1700 FORI=1TO10
1710 N$=NL$(I)
1720 FORJ=1TOLEN(N$):A$=MID$(N$,J,1):PRINTA$;:IFA$=" "T
HEN1740
1730 NEXTJ
1740 PRINT:PRINTSPC(10);CHR$(11);PL$(I):PRINTSPC(25);CH
R$(11);PL$(D
(I))
1750 NEXTI
1770 PRINT:PRINT"    You got ";T;" right"
1780 PRINT:PRINT"    Your final score is ";SC+2*T
5000 DATA16
5010 DATAJohn Smith,Train Set
5020 DATAHarry Todd,Model Car
5030 DATADavid Carter,Bicycle
5040 DATARon Brown,Toy Fort
5050 DATACarol Daly,Fluffy Bunny
5060 DAT~Claire Roberts,Doll
5070 DATASylvia Black,Doll's Pram
5080 DATAAileen Gray,Noddy Book
5090 DATAHenry Forrest,Video Games
5100 DATAWilliam Batey,Scalextrix
5110 DATAKevin Vincent,Big Trak
5120 DATACharlie Knight,Home Computer
5130 DATAJackie Drummond,Typewriter
5140 DATAFrancis White,Tea Set
5150 DATAMarilyn Jones,Doll's House
5160 DATAKathy Morgan,Painting Set
```

Lawrence Lynch

MISSILE ATTACK

You are in charge of the combat computer engaged in saving cities from death and destruction.

You are hidden deep underground manning your combat computer. The alarm sounds and your screen flashes up with an imminent air raid warning. Switching quickly to defence mode, you are presented with a graphic simulation of the three cities it is your duty to defend.

As the enemy missiles descend towards the cities you must intercept them with your automatic range-finder and release a destructive laser bolt. Surviving three attacks (3000 bonus) will earn you 5 laser bolts to replenish your dwindling reserves.

Surviving six attacks (6000 bonus) will promote you to defend three new cities (ie replacing any lost) and reset your lasers to maximum (20 bolts).

WARNING At any moment a missile may release an aerial

HOW IT RUNS

Line 0	Gosub Instructions.
Line 3	Gosub redefined characters.
Line 6	Initialise variables.
Lines 25-250	Set start co-ordinates for missiles.
Lines 300-750	Main Loop.
Line 420	Z=Random aerial bombs.
Line 460	Checks if aerial bomb is over city.
Line 640	Sets gunsight.
Lines 660-690	Allow gunsight to disappear off one side of the screen to re-appear opposite.
Line 770	Increments incoming missiles.
Lines 900-920	Explode missile when hit.
Lines 1000-1210	Explose city if missile hits.
Lines 1500-1550	Erase laser explosions.
Lines 1900-2450	Move gunsight.
Lines 4000-4350	Fire laser.
Lines 4990-5390	Draw landscape and delete cities hit.
Lines 6000-8030	Explode city if hit by bomb.
Lines 10000-10030	Place text on HI RES screen.
Lines 11000-12000	End game or replay: HS=Hi-score.
Lines 12000-12085	Instructions.
Lines 13000-14000	Hi-score and music.
Lines 14000-14090	Low score music.
Lines 14090-16000	Redefining graphics.
Lines 16000-16160	Set colour.

bomb which will destroy a city if it explodes over it. It should also be noted that should the gunsight be moved off the edge of the screen it will re-appear on the opposite side.

HINTS ON CONVERSION

Apart from the music and sound commands and screen co-ordinates which will have to be altered to suit other micros the Oric uses microsoft BASIC which should make it relatively easy to convert.

As the program was written on a 48K Oric, 16K Oric owners will have to change the POKES in Lines 15000 & 15020 from 46080 to 13312. Other micro owners wishing to convert the program will have to redefine their own graphics.

VARIABLES USED

C1	Cities.
C1	City 1
C2	City 2
C3	City 3
HS	H1-Score.
S	Score.
R	Bonus Points Z1
Z1	No. of missiles destroyed.
L	No. of missiles on screen.
X,Y	Co-ordinates of Gunsight.
L(J), Y1(J), KL(J)	Co-ordinates of missiles.
Z	Generate random aerial bombs.
MM	Detects when laser fired.

```
0 PRINTCHR$(6)%CLS:PAPER4:INK3:GOSUB1200
0
3 GOSUB15000
6 CITY=3:RT=20:L=2:Z1=0:C1=0:C2=0:C3=0:S
=0:R=0
10 HIRES: PRINTCHR$(17):IFL>4THENL=4
12 IFRT>20THENRT=20
15 CLS:PRINT"LASER BOLTS LEFT "RT"     SC
ORE "S
16 PRINT"BONUS POINTS "R"     HI SCORE "H
S
20 GOSUB5000
25 REM***SET MISSILES ***
50 X=120:Y=100
100 FORJ=1TOL:L(J)=INT(RND(1)*210)+13
150 Y1(J)=INT(RND(1)*30)+8:KL(J)=1
250 CURSETL(J),Y1(J),KL(J):DRAW2,2,1:NEX
TJ
280 REM*** MAIN LOOP ***
300 REPEAT
350 FORJ=1TOL:Y1(J)=Y1(J)+3:L(J)=L(J)+3
355 IFKL(J)=0THENZ=0:GOTO650
357 IFMM=1THENGOTO360
359 CURSETX,Y,3:CHAR98,0,0
360 IFL(J)=>228THENL(J)=19
370 IFL(J)>=36ANDL(J)<=70ANDY1(J)=>157TH
ENGOTO1000
380 IFL(J)=>106ANDL(J)=<140ANDY1(J)=>157
THENGOTO1100
390 IFL(J)=>172ANDL(J)=<206ANDY1(J)=>157
THENGOTO1200
400 IFPOINT(L(J),Y1(J))=-1THENEXPLODE:Z1
=Z1+1:GOSUB900:KL(J)=0
420 IFY1(J)>40ANDY1(J)<167THENZ=INT(RND(
1)*100)
455 IFZ=5ORZ=20THENFORA=1TO2:CURSETL(J),
Y1(J),3:DRAW0,167-Y1(J),2:N
EXTA
456 IFZ=5ORZ=20THENZAP:EXPLODE:GOSUB900
460 IFZ=5 OR Z=20THENGOTO6000
490 REM*** MOVE MISSILES ***
500 CURSETL(J),Y1(J),KL(J):Z=0:DRAW2,2,1
550 IFY1(J)=>186THENEXPLODE:GOSUB900:KL(
J)=0:Z1=Z1+1
640 CURSETX,Y,3:CHAR98,0,1
650 NEXTJ:GOSUB1500
660 IFX=<20THENX=224
670 IFY=>140THENY=140
680 IFX=>225THENX=21
690 IFY=<18THENY=18
700 IFCI<2THENGOTO807
750 UNTILZ1=>L
760 Z1=0:SS$="BONUS1000":CURSET70,95,3:G
OSUB10000:S=S+1000:WAIT100
770 L=L+1:PRINTCHR$(17):R=R+1000
780 IFR=3000THENRT=RT+5
790 CLS:PRINT"LASER BOLTS LEFT "RT"     S
CORE "S
795 PRINT"BONUS POINTS "R"     HI SCORE "
HS
800 IFR=6000THENC1=0:C2=0:C3=0:RT=20:GOS
UB17000:WAIT300
806 GOTO10
807 SS$="YOU MUST NOW SURRENDER":CURSET4
3,100,3:GOSUB10000:WAIT200
:GOSUB14000
808 GOTO11000
810 PRINTCHR$(6):PAPER7:INK0:CLS:END
900 FORW=1TO6:CIRCLEW,1:NEXTW:FORW=1TO8:
CIRCLEW,0:NEXTW
905 IFZ=5ORZ=20THENRETURN
910 IFY1(J)<148THENS=S+100:CLS:PRINT"LAS
ER BOLTS LEFT "RT"     SCORE
 "S
```

```
915 IFY1(J)<148THEN PRINT"BONUS POINTS "
R"    HI SCORE "HS
920 RETURN
1000 IFC1=1THENGOTO500
1001 C1=1:CITY=CITY-1:Z1=Z1+1:EXPLODE:KL
(J)=0
1005 C=1:FORA=1TO2:FORP=1TO25:CURSET54,1
74,0:CIRCLEP,C:NEXTP:C=0:NE
XTA
1010 GOTO650
1100 IFC2=1THENGOTO500
1101 C2=1:CITY=CITY-1:Z1=Z1+1:EXPLODE:KL
(J)=0
1105 C=1:FORA=1TO2:FORP=1TO25:CURSET126,
174,0:CIRCLEP,C:NEXTP:C=0:N
EXTA
1110 GOTO650
1200 IFC3=1THENGOTO500
1201 C3=1:CITY=CITY-1:Z1=Z1+1:EXPLODE:KL
(J)=0
1205 C=1:FORA=1TO2:FORP=1TO25:CURSET189,
174,0:CIRCLEP,C:NEXTP:C=0:N
EXTA
1210 GOTO650
1500 REM***ERASE EXPLOSION***
1505 IFMM=1THEN1510ELSE2000
1510 A=0:B=199:C=A+X+2:D=Y-B+3:MM=0
1520 CURSET0,199,0
1530 DRAWC,D,3
1540 FORA=1TO12:CIRCLEA,0
1550 NEXTA
1900 REM**SET AND MOVE GUNSIGHT***
2000 REM
2110 CURSETX,Y,3:CHAR98,0,1
2130 K$=KEY$:IFK$=""THENRETURN
2150 IFK$=CHR$(32)THENGOTO4000
2300 IFK$=CHR$(8)THENCHAR98,0,0:X=X-9:RE
TURN
2350 IFK$=CHR$(9)THENCHAR98,0,0:X=X+9:RE
TURN
2400 IFK$=CHR$(10)THENCHAR98,0,0:Y=Y+9:R
ETURN
2450 IFK$=CHR$(11)THENCHAR98,0,0:Y=Y-9:R
ETURN
3600 REM***FIRE LASER***
4000 IFRT=0THENRETURN
4005 A=0:B=199:MM=1:RT=RT-1
4010 CLS:PRINT"LASER BOLTS LEFT "RT"
SCORE "S
4020 PRINT"BONUS POINTS "R"    HI SCORE
"HS
4050 C=A+X+2:D=Y-B+3
4100 CURSET0,199,0:DRAWC,D,2
4150 CURSET0,199,0:DRAWC,D,2
4250 SHOOT
4300 FORA=1TO12:CIRCLEA,1
4350 NEXTA:RETURN
4990 REM *** LANDSCAPE ***
5000 FORX=187TO199
5001 CURSET0,X,1:DRAW239,0,1:NEXTX
```

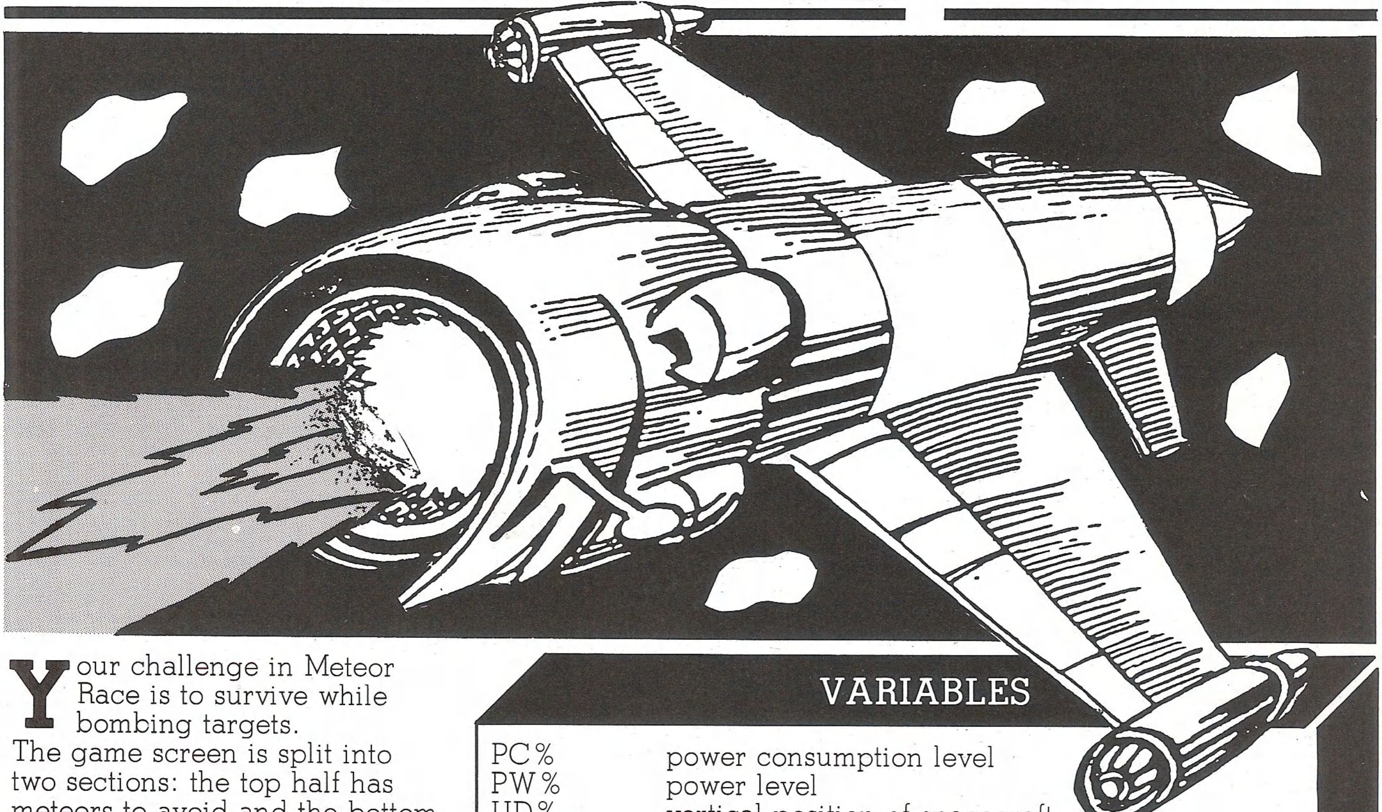
```
5005 GOSUB16000
5008 REM *** MOUNTAIN ***
5009 X=13:Y=190:A=26:REPEAT
5010 CURSETX,Y,1:DRAWA,0,1
5011 X=X+1:Y=Y-1:A=A-2
5012 UNTILA=<0:IFC1=1THENGOTO5060
5020 SS$="ccccc":CURSET36,183,0:GOSUB100
00
5030 SS$="cc cc":CURSET36,175,0:GOSUB100
00
5040 SS$="cc  c":CURSET36,167,0:GOSUB100
00
5050 SS$="c   c":CURSET36,159,0:GOSUB100
00:CURMOV-5,0,1:DRAW0,-6,1
5060 X=72:Y=190:A=30:REPEAT
5070 CURSETX,Y,1:DRAWA,0,1
5080 X=X+1:Y=Y-1:A=A-2
5090 UNTILA=<0
5220 CURSET74,1,3
5230 SS$="MISSILE ATTACK":GOSUB10000:IFC
2=1THENGOTO5280
5240 SS$="ccccc":CURSET106,183,0:GOSUB10
000
5250 SS$="cc cc":CURSET106,175,0:GOSUB10
000
5260 SS$="cc  c":CURSET106,167,0:GOSUB10
000
5270 SS$="c   c":CURSET106,159,0:GOSUB10
000:CURMOV-5,0,1:DRAW0,-6,1
5280 X=140:Y=190:A=30:REPEAT
5290 CURSETX,Y,1:DRAWA,0,1
5300 X=X+1:Y=Y-1:A=A-2
5310 UNTILA=<0:IFC3=1THENGOTO5360
5320 SS$="ccccc":CURSET172,183,0:GOSUB10
000
5330 SS$="cc cc":CURSET172,175,0
:GOSUB10
000
5340 SS$="cc  c":CURSET172,167,0:GOSUB10
000
5350 SS$="c   c":CURSET172,159,0:GOSUB10
000:CURMOV-5,0,1:DRAW0,-6,1
5360 X=206:Y=190:A=29:REPEAT
5370 CURSETX,Y,1:DRAWA,0,1
5380 X=X+1:Y=Y-1:A=A-2
5390 UNTILA=<0:RETURN
6000 IFL(J)>36 AND L(J)<70THENGOTO6001EL
SEGOTO7000
6001 IFC1=1THENGOTO500
6002 C1=1:CITY=CITY-1:EXPLODE
6005 C=1:FORA=1TO2:FORP=1TO25:CURSET54,1
74,3:CIRCLEP,C:NEXTP:C=0:NE
XTA
6010 GOTO490
7000 IFL(J)>106 AND L(J)<140THENGOTO7001
ELSEGOTO8000
7001 IFC2=1THENGOTO500
7020 C2=1:CITY=CITY-1:EXPLODE
7100 C=1:FORA=1TO2:FORP=1TO25:CURSET126,
174,3:CIRCLEP,C:NEXTP:C=0:N
EXTA
```

```
7110 GOTO490
8000 IFL(J)>172ANDL(J)<206THENGOTO8001EL
SEGOTO500
8001 IFC3=1THENGOTO500
8010 C3=1:CITY=CITY-1:EXPLODE
8020 C=1:FORA=1TO2:FORP=1TO25:CURSET189,
174,0:CIRCLEP,C:NEXTP:C=0:N
EXTA
8030 GOTO490
9999 REM*** PLACE TEXT ***
10000 FORI=1TOLEN(SS$)
10010 CHAR ASC(MID$(SS$,I,1)),0,1
10020 CURMOV7,0,3
10030 NEXTI:RETURN
11000 TEXT:PRINT
11005 PRINT:PRINT"   YOU HAVE SURRENDERE
D":WAIT200:CLS
11006 IFHS<STHENHS=S:GOTO13000
11020 PRINT:PRINT:PRINT:PRINTSPC(12)CHR$
(4)CHR$(27)"JYOU SCORED "S
11025 PRINT:PRINT:PRINT:PRINTSPC(8)CHR$(
27)"JHIGH SCORE IS "HS;CHR$
(4)
11030 PRINT:PRINT:PRINT:PRINT"  ANOTHER
GO? (Y/N)"
11040 GET A$
11050 IFA$="N"THENGOTO810
11060 IFA$="Y"THENGOTO6
11070 GOTO11040
12000 PRINT:PRINTSPC(14)CHR$(4)CHR$(27)"
NAIR RAID"CHR$(4)
12001 REPEAT
12002 FORX=125TO80STEP-1
12003 SOUND1,X,10:PLAY1,0,0,0
12004 NEXTX:WAIT65:Y=Y+1
12007 UNTILY=4
12008 PLAY0,0,0,0
12009 EXPLODE
12026 PRINT:PRINT:PRINT"MOVE GUNSIGHT AR
OUND SCREEN USING"
12027 PRINT"CURSOR ARROWS ANY OTHER KEY
WILL FIRE"
12028 PRINT"LASER WHICH WILL INTERSECT T
HE GUN"
12029 PRINT"SIGHT AND EXPLODE."
12035 PRINT
12040 PRINT"IF A LASER EXPLOSION TOUCHES
 THE TIP"
12050 PRINT"OF A MISSILE THE MISSILE WIL
L BE"
12060 PRINT"DESTROYED AND SCORE WILL INC
REA⊠ED BY 100 POINTS"
12065 PRINT:PRINT"SURVIVING AN ATTACK WI
TH AT LEAST ONE"
12070 PRINT"CITY INTACT WILL EARN 1000 P
OINTS"
12075 PRINT:PRINT"BEWARE AERIAL BLASTS!!
!
12076 PRINT"=====================
12077 PRINTCHR$(4);
12078 PRINT" "CHR$(27)"JIF TWO CITES FAL
L GAME IS OVER"CHR$(4)
12079 PRINT
12080 PRINT"3000 BONUS POINTS GAINS 5"
12082 PRINT" EXTRA LASER BOLTS (Max 20)"
12083 PRINT"6000 BONUS POINTS GIVES A BO
NUS CITY"
12084 PRINT"AND SETS LASER BOLTS TO 20"
12085 PRINT"Press any key to start"
12090 GETB$:RETURN
13000 PRINT:PRINTSPC(8)CHR$(4)CHR$(27)"N
A NEW HIGH SCORE"
13010 PRINT:PRINT:PRINT:PRINTSPC(10)CHR$
(27)"J"HS;"POINTS";CHR$(4)
13020 A$="1536581:685631"
13030 FORJ=1TOLEN(A$)
13040 IFJ=8THENWAIT50
13050 X=ASC(MID$(A$,J,1))-48
13060 MUSIC1,3,X,0
13070 MUSIC2,4,X,0
13075 MUSIC3,5,X,0
13080 PLAY7,0,5,400:WAIT50
13090 NEXTJ
13100 WAIT150:PLAY0,0,0,0
13130 GOTO11030
13900 REM*** FUNERAL MARCH ***
14000 IFS>HSTHENRETURN
14003 A$="66669886656"
14005 FORJ=1TO11
14009 X=ASC(MID$(A$,J,1))-48
14010 IFJ=2ORJ=3ORJ=5THENWAIT40
14012 IFJ=6ORJ=8ORJ=10THENWAIT40
14030 MUSIC2,2,X,0
14040 MUSIC3,3,X,0
14050 PLAY6,0,5,90:WAIT10
14060 NEXTJ
14070 WAIT100:PLAY0,0,0,0
14090 RETURN
14900 REM*** DEFINE GRAPHICS ***
15000 FORA=0TO7:READD:POKE46080+A+(8*ASC
("b")),D:NEXTA
15010 DATA0,54,34,8,34,54,0,0
15020 FORA=0TO7:READD:POKE46080+A+(8*ASC
("c")),D:NEXTA
15030 DATA63,45,45,63,45,63,45,63
15060 RETURN
16000 REM *** SET COLOUR ***
16020 CURSET0,0,3:FILL10,1,18
16030 CURSET0,185,3:FILL15,1,18
16040 CURSET6,187,3:FILL13,1,~
16050 CURSET12,0,3:FILL10,1,0
16120 CURSET0,9,3:FILL178,2,16
16125 CURSET12,9,3:FILL178,1,1
16130 CURSET0,9,3:FILL178,1,18
16140 CURSET234,9,3:FILL178,1,18
16160 RETURN
17000 CURSET19,85,3:SS$="BONUS CITY AND
20 LASER BOLTS!"
17010 GOSUB10000:R=0:CI=3
17020 RETURN
```

A P Gardner

ORIC ACTION

Three great action games for you: Meteor Race, Swerve and Conveyor Belt.



Your challenge in Meteor Race is to survive while bombing targets. The game screen is split into two sections: the top half has meteors to avoid and the bottom has a landscape.

Your ship will appear in the centre of the screen and then the screen will scroll from right to left, achieved by using constantly updated strings.

The landscape section has targets to bomb and to survive you must keep hitting the pods, which increase your power by at least 51 units.

The missiles score points only. You lose units of power for each shot, so don't be too trigger happy!

Before starting you will be asked to choose a difficulty level and this controls your power consumption.

You've lost if you run out of power or crash.

You score 10 points for hitting a pod and 15 points for each missile.

The controls are:

X up
down arrow down
up arrow up

VARIABLES

PC%	power consumption level
PW%	power level
UD%	vertical position of spacecraft
A$	landscape string (not updated)
G$-M$	landscape and meteor string variables
L1$, R1$	variables used in updating string variables
SN	loop counter — used when removing hit objects
PK, K$	keyboard values
SC%	your score
HI	high score

HOW IT WORKS

30	swiches off key-click/cursor and sets screen colours
60-130	scan keyboard, make adjustments, check for collision, PLOT player's square
140-190	adjust computer's position, PLOTs square, checks for an obstacle
200-250	changes computer's direction, checks for collision
260-320	crash routines for you and computer
400-470	PRINT latest scores
500-560	game over, PRINT who won, wait for key-press
5000-5040	automatically adjust position of you and computer
7000-7130	set some variables, PLOT screen
8000-8070	set other variables and select skill level
9000-9110	set up user-defined graphics

SWERVE

You're pitted against the computer in a battle of skill.

You and the computer start at opposite sides of the screen, moving towards each other. Both of you leave a wall behind you and your wall is solid.

You must avoid the walls and smaller obstacles and not leave the game boundary.

The loser is the first to crash five times.

You control your direction with the cursor keys.

HINTS ON CONVERSION

PAPER, INK	control background, foreground colours
WAIT	creates a pause for a specified number of milliseconds
PLOT x,y	is equivalent to PRINT @.
KEY$	scans the keyboard
SCRN (x,y), (screen PEEK)	returns the ASCII value for the position given
CLS	clears the screen
PLAY	creates a sound and can be deleted or replaced.
SHOOT,EXPLODE and PING	are Oric sound commands
SPC	can be replaced by TAB or removed

SWERVE VARIABLES

SL	skill level value
A$, B$	belt strings
LV%	number of remaining lives
LV$	L$ strings to display lives
SC%	score
HI	high score
UD%	vertical position of person
K$	keyboard value
W%	WAIT value — controls speed
C$	stores belt edges

CONVEYOR BELT VARIABLES

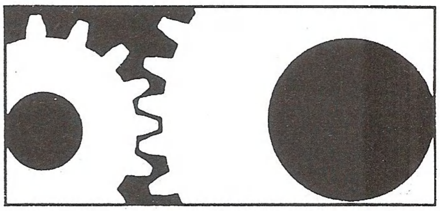
Y	your total score
O	computer's total score
SL	skill level values
P/K$	keyboard values
H%(N), V%(N)	horizontal, vertical positions of computer and you
D%(N)	direction of computer and you
N	controls arrays H%, V% and D%

HOW IT WORKS (SWERVE)

30	switches off key-click/cursor and selects lo-res graphics mode
50-87	select skill level
100-270	set variables
310-385	scan keyboard, make adjustments, check for crash, PLOT your craft
420-550	update string variables
600-680	PLOT string variables and power/score levels, check if you have run out of power
700	loop back to reset string variables (and so replaces targets hit)
800-875	check if bomb has hit something
880-960	a target is hit: update score, update string variable to remove hit object
1000-1130	fail routine: PRINTs scores and waits for key-press
1200-1300	set up screen colours
9000-9080	set up user-defined characters

HOW IT WORKS (CONVEYOR BELT)

30	switches off key-click/cursor and selects lo-res graphics mode
60-100	select skill level and make adjustments
107-110	set other variables
130-140	scan keyboard
150-200	adjust belt strings and PLOT them on screen
210-280	check for hit, PLOT person, PLOT latest score
300-384	fail routine, updates lives total and checks for end of game
390-460	end of game, PRINT scores, waits for key
500-550	you've crossed, update score
650-750	set up screen and colours
9000-9050	set up user-defined graphics



CONVEYOR BELT

As an errand person you always get the worst and most dangerous jobs.

The evil wizard has given you a message to take to the manager of the mines. To do this you must cross two lines of conveyor belts which carry lumps of ore.

You must leap from belt to belt without hitting the ore — if you do you're knocked into the machine workings and that can be painful.

The wizard has also cast a spell which makes you start again after you've crossed.

Luckily, you have three lives and you can adjust the level of difficulty.

You move forward by pressing any key and gain five points for every line of belts crossed.

▶

Meteor Race listing.

```
10 REM
20 REM BY A.P.GARDNER
30 PRINTCHR$(17);CHR$(6):LORES0
40 GOSUB9000
45 SC%=0
47 REM CHOOSE DIFICULTY
50 CLS
60 PRINT:PRINT"WHAT POWER CONSUMPTION (1-7)"
65 PRINT:INPUT"7 BEING HARDEST";PC%
70 IFPC%<1THENPC%=1
80 IFPC%>10THENPC%=10
85 PC%=PC%+3
87 CLS
90 GOSUB1200
95 REM SET VARIABLES
100 PW%=500:UD%=7
120 WAIT100
150 A$="ccccccccccccccccccccccccccccccccccccc"
170 FORN=22TO17STEP-1
180 PLOT1,N,A$
190 NEXTN
210 G$=" ccccc f  cccccccc gccccccccccccccccccccccc
ccc cc   fg ccc
 "
220 H$=" f ccc    fcccccc   g ccccccccccccccccccccf
cc c        g
 "
230 I$="   gc       ccccc     g    fccg cccccc     g
c c
 "
240 J$="              cc                   ccc

 "
250 K$="         d         d         d"
260 L$="       d         d         d  "
270 M$="    d         d         d     "
280 REM MAIN LOOP
300 FORC=1TO60
310 PK=PEEK(520)
320 IFPK=180THENUD%=UD%+2
330 IFPK=176ANDUD%>7THENUD%=UD%-2
340 IFPK=156THENPW%=PW%-(PC%*2):GOTO800
360 IFSCRN(14,UD%)<>32ORSCRN(15,UD%)<>32THEN1000
370 PLOT14,UD%,"ab"
385 PLOT16,17,"c"
420 L1$=LEFT$(G$,1):R1$=RIGHT$(G$,59)
430 G$=R1$+L1$
440 L1$=LEFT$(H$,1):R1$=RIGHT$(H$,59)
450 H$=R1$+L1$
460 L1$=LEFT$(I$,1):R1$=RIGHT$(I$,59)
470 I$=R1$+L1$
480 L1$=LEFT$(J$,1):R1$=RIGHT$(J$,59)
490 J$=R1$+L1$
500 L1$=LEFT$(K$,1):R1$=RIGHT$(K$,29)
510 K$=R1$+L1$
520 L1$=LEFT$(L$,1):R1$=RIGHT$(L$,29)
530 L$=R1$+L1$
540 L1$=LEFT$(M$,1):R1$=RIGHT$(M$,29)
550 M$=R1$+L1$
600 PLOT1,16,LEFT$(G$,37)
610 PLOT1,15,LEFT$(H$,37)
620 PLOT1,14,LEFT$(I$,37)
630 PLOT1,13,LEFT$(J$,37)
640 PLOT1,11,K$
650 PLOT1,9,L$
660 PLOT1,7,M$
665 PLOT11,2,"SCORE "+STR$(SC%)
670 PW%=PW%-PC%
675 IFPW%<0THENPLOT12,12,"OUT OF FUEL":GOTO1000
680 PLOT10,24,"POWER "+STR$(PW%)+"   "
690 NEXTC
700 GOTO210
790 REM CHECK FOR HIT
800 FORSN=(UD%+1)TO17
810 IFSCRN(16,SN)<>32THEN850
820 NEXTSN
850 IFSCRN(16,SN)<>99THEN880
860 PLOT16,SN,"e"
870 PLAY0,1,1,2000
875 GOTO360
877 REM SOMETHING IS HIT
880 SC%=SC%+10
890 IFSCRN(16,SN)=102THENPW%=PW%+65:PLOT16,SN,"e":
PLAY0,1,2,1000:GO
TO920
900 SC%=SC%+5:PLOT16,SN,"e":PLAY0,1,1,2000
920 IFSN=14THENI$=LEFT$(I$,15)+" "+RIGHT$(I$,44)
930 IFSN=15THENH$=LEFT$(H$,15)+" "+RIGHT$(H$,44)
940 IFSN=16THENG$=LEFT$(G$,15)+" "+RIGHT$(G$,44)
950 IFSN=17THENF$=LEFT$(F$,15)+" "+RIGHT$(F$,44)
960 GOTO360
990 REM YOU ARE DESTROYED
1000 PLOT14,UD%,"  "
1005 PLAY0,1,1,10000
1010 FORN=7TO0STEP-1
1020 PAPERN
1025 WAIT50
1030 NEXTN
1040 WAIT100
1050 CLS
1060 PRINT:PRINTSPC(11)"YOU'RE DEAD"
1070 PRINT:PRINTSPC(10)"YOU SCORED ";SC%
1080 IFSC%>HITHENHI=SC%
1090 PRINT:PRINTSPC(7)"HIGH SCORE TODAY ";HI
1100 PRINT:PRINTSPC(3)"PRESS [RETURN] TO PLAY AGAIN"
1110 K$=KEY$
1120 IFK$<>""THENIFASC(K$)=13THEN45
1130 GOTO1110
1190 REM SET UP SCREEN COLOURS
1200 PAPER0:INK3
1210 PLOT0,2,6
1220 FORN=7TO11
1230 PLOT0,N,1
1240 NEXTN
1250 FORN=13TO22
1260 PLOT0,N,2
1270 NEXTN
1280 PLOT1,24,6
1300 RETURN
8990 REM USER DEFINED GRAPHICS
9000 FORP=(46080+(97*8))TO(46080+(103*8)+7)
9010 READUS:POKEP,US:NEXTP:RETURN
9020 DATA0,0,48,63,63,3,0,0
9030 DATA0,0,24,36,63,32,0,0
9040 DATA63,63,63,63,63,63,63,63
9050 DATA0,30,31,63,63,63,24,0
9060 DATA18,9,36,18,9,36,18,9
9070 DATA30,18,18,18,18,12,18,33
9080 DATA12,12,12,30,12,12,30,30
```

Swerve listing.

```
10 REM SWERVE
20 REM BY A.P.GARDNER
30 PRINTCHR$(17);CHR$(6):PAPER0:INK3
40 GOSUB9000
50 GOSUB8000
55 GOSUB7000
57 REM MAIN LOOP
60 P=PEEK(520):N=0
70 IFP=172THEND%(N)=1
80 IFP=180THEND%(N)=2
90 IFP=188THEND%(N)=3
100 IFP=156THEND%(N)=4
110 GOSUB5000
120 IFSCRN(H%(N),V%(N))<>32THEN300
130 PLOTH%(N),V%(N),"a"
140 N=1
150 GOSUB5000
160 IFSCRN(H%(N),V%(N))<>32THEN200
170 PLOTH%(N),V%(N),"b"
180 H1%=H%(N):V1%=V%(N)
190 GOTO60
195 REM ADJUST COMPUTER'S DIRECTION
200 F=INT(RND(1)*2):H%(N)=H1%:V%(N)=V1%
210 IFF=0THEND%(N)=D%(N)-1
220 IFF=1THEND%(N)=D%(N)+1
223 IFD%(N)<1THEND%(N)=4
226 IFD%(N)>4THEND%(N)=1
230 GOSUB5000
240 IFSCRN(H%(N),V%(N))=32THEN170
241 H%(N)=H1%:V%(N)=V1%
243 IFD%(N)=3THEND%(N)=1:GOTO247
244 IFD%(N)=4THEND%(N)=2:GOTO247
245 D%(N)=D%(N)+2
247 GOSUB5000
250 IFSCRN(H%(N),V%(N))=32THEN170
```

```
255 REM COMPUTER CRASH
260 SHOOT
270 FORN=7TO0STEP-1:PAPERN:WAIT20:NEXTN
280 Y=Y+1
290 GOTO400
295 REM YOU CRASH
300 EXPLODE
310 FORN=7TO0STEP-1:PAPERN:WAIT20:NEXTN
320 O=O+1
390 REM SHOW LATEST SCORES
400 WAIT100
410 CLS
420 IFY=5THEN510
430 IFO=5THEN500
440 PRINT:PRINT:PRINTSPC(11)"SCORES SO FAR"
450 PRINT:PRINTSPC(10)"ORIC-1 ";O;" YOU ";Y
460 WAIT300
470 GOTO55
490 REM GAME OVER
500 PRINT:PRINT:PRINTSPC(11)"ORIC-1 WINS":GOTO520
510 PRINT:PRINT:PRINTSPC(13)"YOU WIN"
520 FORN=1TO3:PING:WAIT50:NEXTN
530 K$=KEY$
540 PRINT:PRINT:PRINTSPC(3)"PRESS A KEY FOR ANOTH
ER GAME"
550 K$=KEY$:IFK$=""THEN550
560 GOTO50
4990 REM ADJUSTMENTS TO DIRECTION
5000 IFD%(N)=1THENH%(N)=H%(N)-1
5010 IFD%(N)=2THENV%(N)=V%(N)+1
5020 IFD%(N)=3THENH%(N)=H%(N)+1
5030 IFD%(N)=4THENV%(N)=V%(N)-1
5040 RETURN
6990 REM SCREEN
7000 H%(0)=13:V%(0)=13:D%(0)=3
7010 H%(1)=25:V%(1)=13:D%(1)=1:INK0
7020 PLOT1,0,"cdddddddddddddddddddddddddddddddddd
dde"
7030 FORN=1TO25
7040 PLOT1,N,"i                                 h"
7050 NEXTN
7060 PLOT1,26,"fjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjj
jjjg"
7070 FORN=1TOSL
7080 A=INT(RND(1)*35)+2
7090 B=INT(RND(1)*24)+1
7100 IFB=13THENIFA>12ANDA<26THEN7080
7110 PLOTA,B,"O"
7120 NEXTN:INK3
7130 RETURN
7990 REM VARIABLES, CHOOSE SKILL LEVEL
8000 Y=0:O=0
8010 CLS
8020 PRINT:PRINTSPC(5)"WHAT SKILL LEVEL (1-30)"
8030 PRINT:INPUT"     1 BEING THE EASIEST";SL
8040 IFSL<1THENSL=1
8050 IFSL>30THENSL=30
8060 CLS:WAIT100
8070 RETURN
8990 REM USER DEFINED GRAPHICS
9000 FORP=(46080+(97*8))TO(46080+(106*8)+7)
9010 READU:POKEP,U:NEXTP:RETURN
9020 DATA63,63,63,63,63,63,63,63
9030 DATA63,33,45,45,45,45,33,63
9040 DATA0,0,0,0,0,0,3,2
9050 DATA0,0,0,0,0,0,63,0
9060 DATA0,0,0,0,0,0,48,16
9070 DATA2,3,0,0,0,0,0,0
9080 DATA16,48,0,0,0,0,0,0
9090 DATA16,16,16,16,16,16,16,16
9100 DATA2,2,2,2,2,2,2,2
9110 DATA0,63,0,0,0,0,0,0
```

Conveyor Belt listing.

```
10 REM CONVEYER-BELT
20 REM BY A.P.GARDNER
30 PRINTCHR$(17);CHR$(6):LORES0
35 PAPER0:INK3
40 GOSUB9000
50 CLS
60 PRINT:PRINT:INPUT"WHAT SKILL LEVEL (1-4)";SL
63 IFSL<1THENSL=1
67 IFSL>4THENSL=4
70 IFSL=1THENA$="   ccd       d       cd    cdc"
80 IFSL=2THENA$="     ccd      d      cd  cd    c"
90 IFSL=3THENA$="  ccd     d     cd  cdcc    cdd"
100 IFSL=4THENA$="   ccd    d   cd    cdcc  c  cdd"
105 LV$=" aa"
107 C$="------------------------------------"
110 SC%=0:UD%=25:B$=RIGHT$(A$,34)+LEFT$(A$,2)
:LV%=3:L$= LEFT$(LV$,LV
%):W%=1
115 IFSL>1THENW%=SL*5
120 GOSUB650
130 K$=KEY$
140 IFK$<>""THENUD%=UD%-2
150 R$=RIGHT$(A$,35):L$=LEFT$(A$,1)
160 A$=R$+L$
170 R$=RIGHT$(B$,1):L$=LEFT$(B$,35)
180 B$=R$+L$
190 PLOT1,7,A$:PLOT1,9,B$:PLOT1,11,A$:PLOT1,13,B$
200 PLOT1,17,A$:PLOT1,19,B$:PLOT1,21,A$:PLOT1,23,B$
210 PLOT18,25," ":PLOT18,15," "
220 IFSCRN(18,UD%)<>32THEN300
230 PLOT18,UD%,"a"
240 IFUD%<7THEN500
250 PLOT7,1,STR$(SC%):PLOT7,2,STR$(SC%)
260 IFSC%>HITHENHI=SC%
270 PLOT23,1,STR$(HI):PLOT23,2,STR$(HI)
275 WAITW%
280 GOTO130
290 REM YOU'RE HIT
300 PLOT18,UD%,CHR$(126)
310 PLAY0,1,1,500
315 IFUD%<17THENSC%=SC%+5
320 WAIT50
330 PLOT18,UD%,"b"
340 PLAY0,1,1,5000
350 PAPER7:WAIT50
360 PAPER1:WAIT50
370 PAPER0:WAIT100
380 PLOT18,UD%," "
381 LV%=LV%-1
382 L$=LEFT$(LV$,LV%)+" "
383 PLOT34,1,L$:PLOT34,2,L$
384 IFLV%>0THENUD%=25:K$=KEY$:GOTO150
390 CLS
400 PRINT:PRINTSPC(11)"YOU'RE DEAD"
410 PRINT:PRINTSPC(10)"YOU SCORED ";SC%
415 IFSC%>HITHENHI=SC%
420 PRINT:PRINTSPC(10)"HIGH SCORE ";HI
430 PRINT:PRINTSPC(3)"PRESS [RETURN] TO PLAY AGAIN"
440 K$=KEY$
450 IFK$<>""THENIFASC(K$)=13THENGOTO50
460 GOTO440
490 REM YOU'VE CROSSED!
500 FORN=1TO3:PLAY0,1,2,100:WAIT50:NEXTN
520 PLOT18,UD%," "
530 UD%=25:SC%=SC%+10
540 PLOT18,UD%,"a"
550 GOTO130
640 REM SCREEN
650 CLS
660 POKE48081,10:POKE48121,10
670 PLOT2,1,"SCORE":PLOT2,2,"SCORE"
680 PLOT29,1,"LIVES":PLOT29,2,"LIVES"
681 PLOT34,1,L$:PLOT34,2,L$
682 PLOT15,1,"HI-SCORE":PLOT15,2,"HI-SCORE"
685 PLOT1,1,2:PLOT1,2,2
690 PLOT1,7,A$:PLOT1,9,B$:PLOT1,11,A$:PLOT1,13,B$
695 PLOT1,6,C$:PLOT1,8,C$:PLOT1,10,C$:PLOT1,12,C$:
PLOT1 ,14,C$
700 PLOT1,17,A$:PLOT1,19,B$:PLOT1,21,A$:PLOT1,23,B$
705 PLOT1,16,C$:PLOT1,18,C$:PLOT1,20,C$:PLOT1,22,
C$:PLO T1,24,C$
710 PLOT18,25,"a"
720 FORN=7TO23STEP2
730 PLOT0,N,1
735 NEXTN
740 PLOT0,5,2:PLOT0,15,2:PLOT0,25,2
750 RETURN
8990 REM USER DEFINED GRAPHICS
9000 FORP=(46080+(97*8))TO(46080+(100*8)+7)
9010 READUS:POKEP,US:NEXTP:RETURN
9020 DATA31,17,31,4,31,4,10,17
9030 DATA33,0,0,12,12,0,0,33
9040 DATA0,12,14,30,30,28,8,0
9050 DATA0,0,4,12,12,0,0,0
```

Sean Rogers, Simon Overy

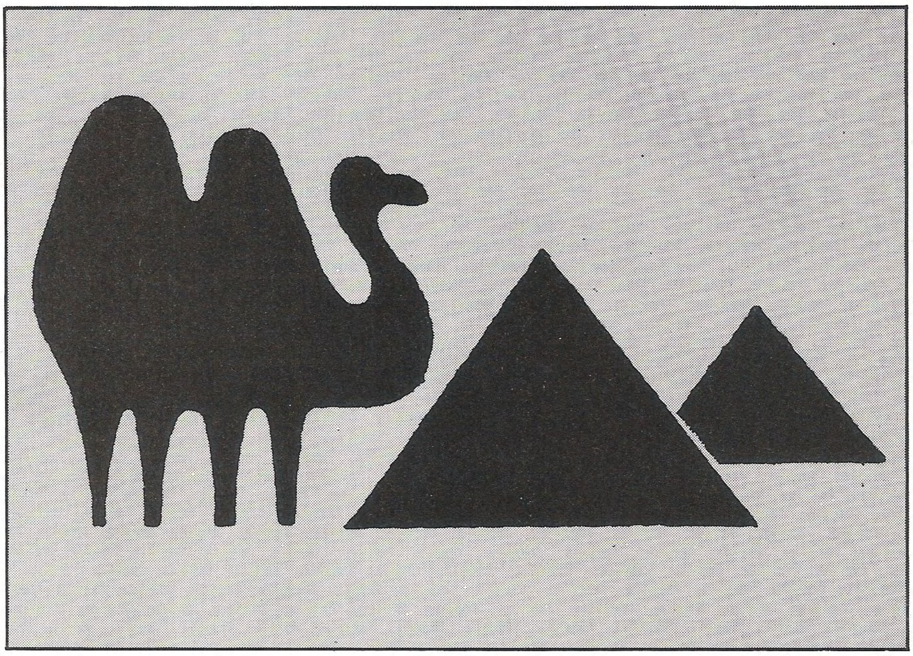
PYRAMID AND ELECTRIC EEL

Misfortunes and shocks are in store for you with these two games for the Oric.

You have just had the misfortune of being transported to the inside of a pyramid from which there seems to be no escape.
You must run around and collect the required number of keys and then make your way to the top to get a part of the Lebasi treasure.

Do this as fast as possible and you'll get a good bonus.

By the way, watch out for the monster!



MAIN VARIABLES

X,Y	your position
BO	bonus
LI	lives
SC	score
MX,MY	monster's position

Movement is by the cursor keys.

The pyramid might look a bit like a Christmas tree, but I needed the space.

When your bonus drops below 1,000 the fourth digit will stay at 0 and so on. This will make it harder to notice when your bonus is low.

HINTS ON CONVERSION

PLOT in Oric BASIC is the same as PRINT AT. The Oric uses an 8x6 grid for user-defined graphics compared with the 8x8 of most other computers.

HOW IT WORKS

10-27	set colour, turn off caps, cursor and key-click, reset DATA pointer
30-40	branch to user-defined graphics and instructions
45	sets variables
47-317	set up screen
318-420	move you around screen, check for keys, plot you on screen
430-530	move monster and check if you have been killed
1800-1940	routine for when bonus runs out and for when the monster gets you
2000-2090	check if you have enough keys if you make it to the top.
2100-2140	message when all lives lost
2200-2260	music
2600-2630	routine to see how many keys are needed
3000-3170	user-defined graphics
3180-3190	more variables
3540-3630	instructions

SLIPPERY EEL

You are an eel in a large aquarium and to stay alive you must swim about eating one kind of fish but avoiding the other type.

HOW IT WORKS

20-25	title
160-330	instructions
340-360	tune
4000-4080	end of game
5000-5910	main game routine
6000-6170	display start screen
7000-7010	set game variables
9000-9530	initialisation and user-defined graphics

A loud continuous noise tells you that you are short of oxygen and must swim to the surface within a certain time.

The deeper the Mouldfish are the more points you will score for eating them. The eel increases its length every time it eats something.

The game ends when either your oxygen runs out, you are hit — or touched horizontally — bu a Spikefish, or there are no Mouldfish left.

VARIABLES

SC	present score
HI	high score
OX	indicates oxygen needed
XC	counter for oxygen left
EP(I,J)	position of eel
EF,EB	front, back of eel
MF(I,J)	position of Mouldfish
MC	counter for Mouldfish left
EX,EY	direction of eel
T(I)	tune array

HINTS ON CONVERSION

Leave out all the POKEs and use your own computer's clock. Omit double height title (CHR$s 4 and 27) and change 6000-6170 to suit your own screen.

POKE 48035,7 turns CAPS off.
POKE (hash) 26A,10 turns off key-click and cursor
DOKE 630,n contols internal timer
CHR$s 91, 92 and 93 are the user graphics.

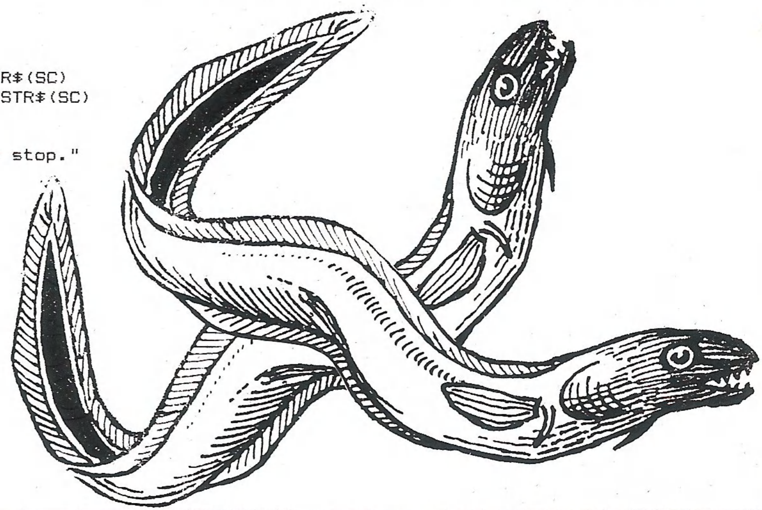
The listing for Pyramid.

```
0 REM          ***********
1 REM         *************
2 REM        ***************
3 REM       *****************
4 REM     **/P Y R A M I D\**
5 REM    *********************
6 REM   ***********************
7 REM *************************
8 REM*           S.ROGERS        *
9 REM***************************
10 PAPER0:INK7:CLS
20 PRINTCHR$(20);CHR$(17);CHR$(6)
27 RESTORE
30 GOSUB 3000
40 GOSUB 3540
45 X=4:Y=25:MX=23:MY=8:BO=8000
47 CLS
50 PLOT20,2,"lk"
60 PLOT19,3,"ljjk"
70 PLOT18,4,"l    k"
80 PLOT17,5,"lj %& jk"
90 PLOT17,6,"lj    jk"
100 PLOT16,7,"ljj  jjjjk"
110 PLOT16,8,"l        jk"
120 PLOT15,9,"lj        jjk"
130 PLOT14,10,"ljjjjjjj  jjjk"
140 PLOT13,11,"lj    jjj    i k"
150 PLOT12,12,"lj     jjj        k"
160 PLOT12,13,"lj  j        jjj  k"
170 PLOT11,14,"ljj  j        jjj  jk"
180 PLOT11,15,"ljj ijjjjjj  jjj   k"
190 PLOT10,16,"ljjj  jjjjjj  jjj   jk"
200 PLOT10,17,"ljjjjjjjjjjj  jjjj  jk"
210 PLOT9,18,"ljj                ijj   jk"
220 PLOT8,19,"ljjj                jji     k"
230 PLOT7,20,"lj      jjjjjjjjj  jj      jk"
240 PLOT6,21,"ljji     jjjjjjjjj  jjjjjj jjk"
250 PLOT5,22,"ljjjjjj  j                  j  jjk"
260 PLOT4,23,"ljjjjjjj  ji                 j  j ik"
270 PLOT0,24,"jjjjjjjjjjjjjj  jjjjjjjjj  jjj      jk"
280 PLOT26,25,"j            jj"
290 PLOT0,26,"jjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjj"
300 PLOT27,0,"BONUS"
305 PLOT27,1,"SCORE"
310 PLOT27,2,"LIVES"
312 PLOT32,0,STR$(BO)
315 PLOT32,1,STR$(SC)
317 PLOT32,2,STR$(LI):K$=KEY$
318 IF BO=0 THEN 1800 ELSE BO=BO-25
319 PLOTX,Y," "
320 IFSCRN(X-1,Y)=105 THEN SC=SC+150:D=D-1:X=X-1:GOTO4
00
330 IFK$=CHR$(8) AND SCRN(X-1,Y)<106 THEN X=X-1:GOTO40
0
340 IFSCRN(X+1,Y)=105 THEN SC=SC+150:D=D-1:X=X+1:GOTO4
00
350 IFK$=CHR$(9) AND SCRN(X+1,Y)<106 THEN X=X+1:GOTO40
0
360 IFSCRN(X,Y+1)=105 THEN SC=SC+150:D=D-1:Y=Y+1:GOTO4
20
370 IFK$=CHR$(10)AND SCRN(X,Y+1)<106 THEN Y=Y+1:GOTO40
0
380 IFSCRN(X,Y-1)=105 THEN SC=SC+150:D=D-1:Y=Y-1:GOTO4
00
390 IFK$=CHR$(11)AND SCRN(X,Y-1)<106 THEN Y=Y-1
400 IFY=7THEN GOSUB 2000
410 IFX<=1 THEN X=2
420 PLOTX,Y,"h"
430 PLOTMX,MY," "
450 IFMY<Y AND SCRN(MX,MY+1)<106THEN MY=MY+1:GOTO510
460 IFMX<X AND SCRN(MX+1,MY)<106THEN MX=MX+1:GOTO490
470 IFMY>Y AND SCRN(MX,MY-1)<106THEN MY=MY-1:GOTO490
480 IFMX>X AND SCRN(MX-1,MY)<106THEN MX=MX-1
490 IFMX<=1 THEN MX=2
510 PLOTMX,MY,"!"
520 IFSCRN(X,Y)=33 THEN 1900
530 GOTO312
1800 PLOTX,Y,"m":GOSUB2200
1810 LI=LI-1:IFLI<=0THEN 2100
1820 CLS:PLOT9,13,"YOUR BONUS HAS RUN OUT":WAIT50
1830 PLOT11,15,N$:WAIT90
1835 GOSUB2600
1840 GOTO45
1900 PLOTX,Y,"m":GOSUB2200
1910 LI=LI-1:IFLI<=0THEN 2100
1920 CLS:PLOT9 ,13,"THE MONSTER GOT YOU":WAIT50
1930 PLOT11,15,N$:WAIT90
1935 GOSUB2600
1940 GOTO45
2000 IFD<=0 THEN 2020
2010 Y=8:RETURN
2020 IF LI<5 THEN LI=LI+1
2030 L=L+1:SC=SC+BO:GOSUB 2600
2035 CLS
2040 PLOT9,13,CHR$(142)+" CONGRATULATIONS"
2050 PLOT9,14,CHR$(142)+" CONGRATULATIONS"
2060 PRINT:PRINT"YOU HAVE GOT "L" PARTS OF THE LEBASI"
:PRINT
2070 PRINTSPC(14)"TREASURE":WAIT200
2080 PRINT:PRINTSPC(11)N$:WAIT100
2090 GOTO45
2100 CLS
2105 PRINT"YOU HAVE NO MORE LIVES LEFT":PRINT
2110 PRINT"YOUR SCORE IS "SC:PRINT
2120 PRINT"YOU HAVE "L" PART(S) OF THE LEBASI TREASURE
"
2130 PRINT:PRINT"GET READY FOR ANOTHER GAME"
2140 WAIT 2000:GOTO 27
2200 FORT=12 TO 1 STEP-1:WAIT10
2230 MUSIC1,6,T,10
2240 NEXTT:PING
2250 WAIT100
2260 RETURN
2600 IF SC>8000 THEN D=5 ELSE D=4
2610 IF SC>16000 THEN D=6
2620 IF SC>24000 THEN D=7
2630 RETURN
2990 REM GRAPHICS + VARIABLES
3000 FORT=46344 TO 46351:READG:POKET,G:NEXTT
3010 FORT=46376 TO 46391:READG:POKET,G:NEXTT
```

```
3020 FORT=46912 TO 46919:READG:POKET,G:NEXTT
3030 FORT=46920 TO 46927:READG:POKET,G:NEXTT
3040 FORT=46928 TO 46935:READG:POKET,G:NEXTT
3050 FORT=46936 TO 46943:READG:POKET,G:NEXTT
3060 FORT=46944 TO 46951:READG:POKET,G:NEXTT
3070 FORT=46952 TO 46959:READG:POKET,G:NEXTT
3100 DATA 20,42,8,20,28,28,34,34
3110 DATA 0,1,3,15,30,60,62,63,0,0,32,56,60,62,62,62
3120 DATA 12,18,12,15,12,20,34,51
3130 DATA 2,10,14,10,2,2,7,7
3140 DATA 63,63,63,63,63,63,63,63
3150 DATA 32,48,48,56,60,62,62,63
3160 DATA 1,3,3,7,15,31,31,63
3170 DATA 4,4,31,4,4,4,14,31
3180 SC=0:LI=3:D=4
3190 N$="HERE WE GO AGAIN"
3200 RETURN
3540 CLS:PRINTSPC(11)"P Y R A M I D"
3550 PRINT:PRINT:PRINT
3560 PRINT"THE OBJECT OF THE GAME IS TO GET IN":PRINT
3570 PRINT"THE FIRST GAME AT LEAST 4 KEYS FROM":PRINT
3580 PRINT"THEN ON EVERY 8000 POINTS YOU NEED":PRINT
3585 PRINT"ONE EXTRA KEY UNTIL YOU NEED ALL 7.":PRINT
3590 PRINT"YOU THEN GO TO THE ROOM AT THE TOP":PRINT
3600 PRINT"OF THE PYRAMID TO GET A PART OF THE":PRINT
3610 PRINTSPC(11)"LEBASI TRESURE"
3620 PLOT13,26,"PRESS A KEY"
3630 GETA$:RETURN
```

The listing for Electric Eel.

```
 1 REM ELECTRIC EEL by Simon Overy
 10 GOSUB9000
 20 TEXT:CLS:PAPER1:INK0
 21 PRINTSPC(50);CHR$(4);CHR$(27);
 22 PRINT"NELECTRIC EEL";CHR$(4)
 23 SOUND1,37,0:SOUND2,75,0
 24 PLAY3,0,1,4000
 25 PLOT13,10,"By Simon Overy":WAIT300
 40 GOSUB160:GOSUB7000:GOSUB6000
 50 GOSUB5000:GOSUB4000
 90 IFYN$="Y"THEN40
 95 POKE48035,7:CLS:EXPLODE:END
 160 CLS:PAPER3:PRINTSPC(76)
 170 PRINT,"You are an ELECTRIC EEL  "
 180 PRINT,"and must swim around the "
 190 PRINT,"aquarium eating MOULDFISH"
 200 PRINT,"( [ ) but avoiding SPIKE "
 210 PRINT,"FISH( \ ),the glass sides"
 220 PRINT,"and your own tail!       "
 230 PRINT,"   A loud noise indicates"
 240 PRINT,"that you must swim to the"
 250 PRINT,"top before your oxygen   "
 260 PRINT,"runs out.Once you are    "
 270 PRINT,"fully oxygenated you can "
 275 PRINT,"either turn left or right"
 276 PRINT,"You can only surface when"
 280 PRINT,"in need of oxygen .":PRINT
 285 PRINT" HINT:Spikefish have side suction !?!?":PRIN
T
 290 FORI=19TO24:PLOT1,I,21:NEXTI
 300 PLOT2,19,"SCORING:-"
 310 PLOT9,21,"The deeper the better!"
 320 PLOT2,23,"ARROW KEYS TO MOVE"
 330 PLOT1,25,"Hit any key for a swim...":GETK$
 340 FORI=1TO15:SOUND1,T(I)*10,0
 350 SOUND2,T(I)*100,0:SOUND3,T(I)*200,0
 360 PLAY 7,0,1,2000:WAIT35:NEXTI
 370 RETURN
 4000 FORI=100TO500STEP2
 4010 SOUND1,I,10
 4020 NEXTI
 4030 CLS:PAPER5:INK4:PRINTCHR$(4):ZAP
 4040 PRINTSPC(160);CHR$(27);"JS C O R E J ";STR$(SC)
 4050 IFSC>HITHENHI=SC:HI$=RIGHT$(STR$(SC),LEN(STR$(SC)
)-1)
 4060 PRINTCHR$(4)
 4070 PLOT1,20,"Hit 'Y' for play , any other to stop."
 4080 K$=KEY$:GETYN$:RETURN
 5000 DOKE630,1600
 5008 REPEAT
 5015 RR=RND(1)
 5016 IFRR<0.01ANDOX=0THENOX=1:XC=0
 5017 IFOX=1THENGOSUB5800
 5020 SS=SCRN(EP(EF,1),EP(EF,2))
 5030 IFSS=32THENGOSUB5070:GOTO5055
 5040 IFSS=91THENGOSUB5180:GOTO5055
 5052 SS=1:PLOTEP(EF,1),EP(EF,2),93
 5055 UNTIL MC=0 OR SS=1 OR XC=35
 5060 RETURN
 5070 PLOTEP(EF,1),EP(EF,2),93
 5080 PLOTEP(EB,1),EP(EB,2),32
 5090 EF=EF+1:EB=EB+1
 5100 IFEF=401THENEF=1:WP=399:GOTO5125
 5110 IFEB=401THENEB=1
 5120 WP=-1
 5125 K$=KEY$:IFK$<>""THENGOSUB5500
 5130 EP(EF,1)=EP(EF+WP,1)+EX:EP(EF,2)=EP(EF+WP,2)+EY
 5140 IFDEEK(630)>1000THEN5170
 5150 PLOTMF(MC,1),MF(MC,2),32:MC=MC-1
 5160 DOKE630,1600
 5170 RETURN
 5180 PLAY 7,0,1,2000
 5190 SOUND 1,100,0
 5200 SOUND 2,500,0
 5210 SOUND 3,999,0
 5220 SC=SC+(10*EP(EF,2)):GOSUB5900
 5230 FORI=1TOMC
 5240 IFMF(I,1)=EP(EF,1)ANDMF(I,2)=EP(EF,2)THENII=I
 5250 NEXTI
 5260 PLOTMF(II,1),MF(II,2),32
 5270 X=INT(RND(1)*35)+2:Y=INT(RND(1)*22)+3
 5280 IFSCRN(X,Y)<>32THEN5270
 5290 MF(II,1)=X:MF(II,2)=Y:PLOTX,Y,91
 5300 IFEB>3THENEB=EB-3
 5320 II=0:DOKE630,DEEK(630)+350
 5330 RETURN
 5500 K=ASC(K$)
 5510 IFK=8THENEX=-1:EY=0
 5520 IFK=9THENEX=1:EY=0
 5530 IFK=10THENEX=0:EY=1
 5540 IFK=11THENEX=0:EY=-1
 5550 RETURN
 5700 EP(EF,2)=EP(EF,2)+1
 5702 TT=DEEK(630):GETK$:GOSUB5500:DOKE630,TT
 5703 EP(EF,1)=EP(EF,1)+EX
 5705 EY=0:OX=0:XC=0
 5710 RETURN
 5800 IFEP(EF,2)=2THEN5700
 5805 SOUND1,1000,0:SOUND2,300,0:PLAY3,0,1,2000
 5810 XC=XC+1:RETURN
 5900 PLOT10,26,RIGHT$(STR$(SC),LEN(STR$(SC))-1)
 5910 RETURN
 6000 TEXT:CLS:PAPER4:INK2
 6010 FORI=0TO2:FORJ=1TO38:PLOTJ,I,22:NEXTJ,I
 6020 PLOT1,26,23:PLOT2,26,0
 6030 PLOT3,26,"SCORE:"
 6040 PLOT20,26,"HIGH:":PLOT26,26,HI$
 6050 FORI=3TO24
 6060 PLOT1,I,148:PLOT38,I,148
 6070 NEXTI
 6080 FORI=1TO38:PLOTI,25,148:NEXTI
 6090 FORI=1TO10
 6100 X=INT(RND(1)*32)+4:Y=INT(RND(1)*19)+4
 6110 IFSCRN(X-1,Y)<>32ORSCRN(X,Y)<>32ORSCRN(X+1,Y)<>32
THEN6100
 6120 PLOTX-1,Y,SF$
 6130 X=INT(RND(1)*35)+2:Y=INT(RND(1)*19)+4
 6140 IFSCRN(X,Y)<>32THEN6130
 6150 PLOTX,Y,91:MF(I,1)=X:MF(I,2)=Y
 6160 NEXTI
 6170 RETURN
 7000 SC=0:EF=10:EB=1:EX=0:EY=-1:MC=10
 7010 EP(10,1)=2:EP(10,2)=24:RETURN
 9000 DIMEP(400,2),T(15)
 9010 POKE#26A,10:POKE48035,0
 9030 FORI=46808TO46831
 9040 READJ:POKEI,J
 9050 NEXTI
 9060 SF$=CHR$(1)+"\"+CHR$(2)
 9070 FORI=1TO15:READJ:T(I)=J:NEXTI
 9080 RETURN
 9500 DATA 12,12,18,30,33,33,45,18
 9510 DATA 0,18,45,12,12,45,18,0
 9520 DATA 12,18,33,45,45,33,18,12
 9530 DATA 5,15,25,15,8,9,10,5,15,25,15,8,12,16,20
```



Brian and Sarah Grayson

DRAWING THE LINE

Make your Oric quicker on the draw with this nifty little program.

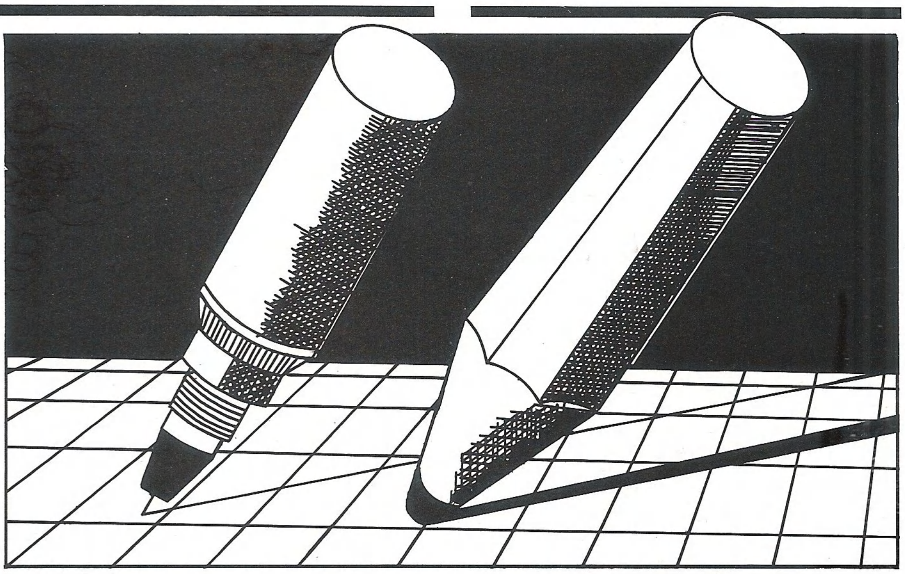
This program enables you to draw lines and circles in the hi-res mode using only the cursor control keys and various assigned single key entries.

No prior plotting on to paper is required as the cursor is used to 'compose' direct on the screen.

The assigned keys are self explanatory and detailed in the program they are:

P	To fix a point to which lines may be drawn from any other cursor position.
C	To draw a circle, you must input a radius when requested.
L	To draw a line to the point previously 'fixed'.
D	To enter the delete mode when every subsequent action is drawm in the background colour (wiping out a previous foreground feature).
N	To return to the normal mode.
Q	Returns to the instructions but beware, all your work will be lost.
S	To save the drawing on the screen. Position a blank piece of tape and start to record before pressing S.

Cursor movement is achieved with the arrow keys, and the distance moved may be varied by 1,2,5 or 10 pixels by pressing 1,2,5,0 at any time. Best control will be achieved by using 5 or 10 until you are near to your desired position, then changing 1 or 2 for final positioning.



HINTS ON CONVERSION

Any machine with reasonably high resolution drawing capabilities could use the program with very ittle modification.

The CIRCLE command, if not built into your machine's BASIC will have to be constructed using thenput radius and sine and cosine.

LINE DRAW may have to be modified depending on the facility to draw relative to the last position or to some new x,y position. Since the Oric cannot draw to a point, those machines with this facility will only require a modification enabling it to draw to V and W.

SAVE will depend on your machine's capability to store screen information. If this can be done, the appropriate SAVE command and memory location will need to be inputted.

DELETE uses the Oric facility to use foreground on background colours. Most machines have a different facility and this part of the program should incorporate the appropriae command to produce an inverse of the normal display.

Most machines use ASCII codes, so the keyboard sensing should not present any difficulties. Only the cursor movement keys will need to be checked.

Remember to place the border slightly inside the full screen size to avoid the program stopping because it is out of range.

HOW IT WORKS

5	Clears the screen and sets up the title sequence and instructions.
10-20	Draws the border, sets the initial cursor position, normal mode, and pixel movement (5).
30-130	Reads the keyboard input and resets the variables or sets up mode.
200-206	Senses if a line is already on the screen to avoid wiping it out as the cursor flashes.
400-602	Title and instructions.
1000-1040	Tape save and reset or program termination.

```
1 REM *** ORIC DRAUGHTSMAN ***
2 REM ** by B.D.& S.L.GRAYSON July '83 *
*
5 CLS:GOTO400
10 HIRES:PRINTCHR$(20);CHR$(17)
15 CURSET0,0,1:DRAW239,0,1:DRAW0,199,1:D
RAW-239,0,1:DRAW0,-199,1
20 X=120:Y=100:T=1:I=5
30 K$=KEY$
32 IFK$=CHR$(49)THENI=1
33 IFK$=CHR$(50)THENI=2
34 IFK$=CHR$(53)THENI=5
35 IFK$=CHR$(48)THENI=10
40 IFK$=CHR$(8)THENX=X-I
50 IFK$=CHR$(9)THENX=X+I
60 IFK$=CHR$(10)THENY=Y+I
70 IFK$=CHR$(11)THENY=Y-I
80 IFK$="P"THENCURSETX,Y,T:GETK$:GOTO32
85 IFK$="Q"THENTEXT:RUN
90 IFK$="C"THENCURSETX,Y,1:PRINT"What ra
dius?":INPUTR:CIRCLER,T:GOT
030
95 IFK$="S"THENGOTO1000
100 IFK$="F"THENCURSETX,Y,1:CIRCLE4,1:CL
S:PRINT"Line draw":V=X:W=Y

110 IFK$="L"THENCURSETV,W,0:CIRCLE4,0:CU
RSETX,Y,1:DRAW(V-X),(W-Y),T

120 IFK$="D"THENPRINT"Delete mode":T=0:G
OTO30
130 IFK$="N"THENCLS:T=1:GOTO30
200 IFPOINT(X,Y)=-1THENCURSETX,Y,0:WAIT5
:CURSETX,Y,1:GOTO30
201 IFPOINT(X,Y)=0THENCURSETX,Y,1:WAIT5:
CURSETX,Y,0:GOTO30
400 POKE618,2:PAPER0:INK7
401 IFPEEK(524)>127THENPRINTCHR$(20)
402 PLOT16,8,20:PLOT17,8,"ORIC":PLOT22,8
,16
404 PLOT12,12,20:PLOT13,12,"DRAUGHTSMAN
" :PLOT25,12,16
450 WAIT175:CLS
455 PRINT:PRINT:PRINT:PRINT
460 PRINT"This Oric program allows you t
o draw"
462 PRINT"lines and shapes in the comput
er's "
464 PRINT       "HIRES mode.
466 PRINT        "Only a few single key co
mmands "
468 PRINT       "are  necessary.":PRINT
469 PRINT       "The finished drawing may
be"
470 PRINT       "saved onto tape and subse
qently"
471 PRINT       "reloaded and displayed, t
hough"
472 PRINT       "not modified."
475 WAIT1500:CLS
500 PRINT"          These are:-":PRINT
506 PRINT"The normal cursor control keys
":PRINT
507 PRINT"P - To fix a point":PRINT
508 PRINT"C - To draw a circle - you mus
t input      a radius":PRINT

509 PRINT"F - To fix a point to which yo
u may"
      510 PRINT"     draw lines from othe
r points":PRINT
                    511 PRINT"L - To draw a
line from the current"
512 PRINT"    position to the 'F' point"
:PRINT
513 PRINT"D - To enter the delete mode."

514 PRINT"    Lines and circles when dra
wn in "
515 PRINT"    the normal way will be era
sed":PRINT
516 PRINT"N - To return to⊠normal mode":
PRINT
517 PRINT"Q - To quit the program and re
turn"
518 PRINT"    to this list. ALL IS LOST!
":PRINT
524 PRINTCHR$(27) "L      Press SPACE to
continue"
525 GETZ$:IFZ$=CHR$(32)THENCLS
528 PRINT:PRINT:PRINT:PRINT:PRINT:
530 PRINT"S - This automatically saves t
he     "
531 PRINT                   "     finished dr
awing on tape"
532 PRINT"    REMEMBER TO POSITION SOME
BLANK"
533 PRINT"    TAPE AND START THE CASSETT
E          BEFORE PRESSING
 'S'"
534 PRINT"    To recover the drawing typ
e"
535 PRINT"    HIRES before 'CLOAD'"
536 PRINT:PRINT:PRINT:PRINT
550 PRINT"  The distance the flashing cu
rsor"
552 PRINT"  moves may be varied at any t
ime"
554 PRINT"  by inputting 1,2,5,0 for1,2,
5,10"
556 PRINT"  pixels.It is normally 5."
590 PRINT:PRINT:PRINT:PRINT:PRINT

⊠600 PRINTCHR$(27)"L        Press SPACE t
o begin"
602 GETZ$:IFZ$=CHR$(32)THENGOTO10
1000 PRINT"saving in progress"
1010 CSAVE"",A40960,E49000,AUTO
1020 CLS:PRINT"Saving complete. Do you w
ish to continue Y/N ?"
1030 INPUTR$:IFR$="Y"THENGOTO10
1040 IFR$="N"THENTEXT:CLS
```

ALIEN SCOUTS

A wave of alien scouts are about to destroy your city; however, you are not totally without defence.

They're back. A new evil race of aliens have taken it upon themselves to wipe out your city. You control the obligatory missile-launcher. With this you must wipe out the alien hordes. Your eyes anxiously look towards the sky, where you expect to see the foe any second now. Suddenly the sky is full of the most horrible, fearful and frightening creatures you have ever seen. The Alien Scouts have arrived....

Firstly a wave Alien Scouts rapidly drop towards the city. If they land the city is doomed. You must act quickly.

Then the Mega-Bombs fall from the immense, unseen Command ship. Some hang suspended by invisible Myocrikle cords, waiting to fall. Others continue to fall, the ever-increasing noise designed to irritate rarely fails. Quick, help! BANG, a few parts of the city are destroyed.

And the third wave...You hear an evil noise, throbbing and threatening. The Saucers hurtle across the screen. A truly awesome sight. Seemingly harmless, their true intent — to supply the others, dawns upon you (usually too late). Although the city isn't likely to be damaged, your defence couldn't survive the onslaught if the extra supplies were received.

If, by a remarkable feat of skill you survive all three waves; you get an extra life and bonus points. Then you must face them again, in a more powerful form.

Alien Scouts should work on a 16K Oric if the command GRAB is used and memory is cleared before starting.

Full instructions are included in the program. There are only four keys to use; left, right, fire, and pause. These can be redefined within the program. The high score is displayed all the time with the name of the holder of it. Five skill levels are already incorporated into the program, but more easily be.

If you find this program too easily beaten, alter the numbers following the variables Y and E in the IF...THEN loops in the second and third waves. Raising

RUNdown

Lines 8-70	Initialisation of control keys. Various jumps to; redefine characters, reset variables, instructions and the game.
Lines 100-110	Control section, where inputs are processed. Movement and pause inputs are handled here. A jump to the firing

	routine is activated by the fire key.
Lines 200-213	The first wave of aliens. Various jumps to: control section, aliens landed section, sheet cleared section, set up display section.
Lines 300-322	The second wave of Aliens, various jumps to: life lost section, control section, set up display, sheet cleared section.
Lines 400-413	The third wave of aliens. Various jumps to: life lost section, sheet cleared section, control section, set up display section.
Lines 1000-1032	Firing routine.
Lines 2000-2090	Life lost section. Checks number of lives left. If you have no lives left, your score is checked against the high score. If this is greater then the high score you are asked to input your name.
Lines 3000-3100	Sheet cleared section. A bonus is calculated and you are given it.
Lines 4000-4130	Aliens landed section. This occurs when an alien scout lands. The game is over and a jump to the life section occurs.
Lines 5000-5340	Set up display section. This produces the display complete with stars, score, high score, name of the high score holder, sheet number and different colour aliens.
Lines 6000-6060	City destroyed section. This occurs when all of the cute red city has been destroyed.
Lines 7000-7120	Cleared three waves section. This section is jumped to when all three waves have been cleared. A bonus is calculated and added to your points. An extra life is awarded. Then a jump to the first screen occurs.
Lines 10000-10230	Instructions section. This requires you to input the skill level and checks if the present keyboard controls are O.K. If not, you have to redefine them. Jumps to the start of the program.
Lines 15000-15260	Redefine characters section.
Lines 20000-20030	Reset variables section. Jumps to start of program.

the variable E increases the number of aliens which you must destroy before going onto the next wave. Lowering the variable Y lowers the number of aliens needed to kill you.

If you find that you cannot enter a line without it overflowing the two lines maximum, then enter the PRINT statements as? (question mark).

CONVERSION CLUES

This program is written entirely in BASIC so that it is easy to convert; and so that it can be modified or even improved.

A few points to remember about the Oric are:
It has a screen size of 38 columns by 26 lines.
There are 4 predefined sound commands; ZAP, SHOOT, EXPLODE and PING. The character is formed in a 6 by 8 grid (not 8 by 8 as in others).
PRINT CHR (17) toggles the cursor on/off (same as CTRL Q).
PRINT CHR (6) toggles the keyboard click on/off (same as CTRL F).
A—FRE (””) forces variable garbage collection.

When a plot command is followed by a number, the number is a serial attribute which controls the screen. Numbers from 0 to 7 alter the foreground colour for that point to the end of that line. Numbers from 8 to 15 change the character set that is printed (for example, the number 12 represents single height flashing characters). Numbers from 16 to 23 alter the background colour for that point to the end of the line.

```
8 REM -------------------------------
9 REM |                             |
10 REM|    A L I E N  S C O U T S   |
11 REM|                             |
12 REM|                             |
13 REM|        By D.T.Wright        |
14 REM|                             |
15 REM-------------------------------
20 CLEAR
30 DIMCD(37):L‡="1":R‡="2":F‡="\":N‡="The Aliens":POKE48035,0:HS‡="
00000"
33 FORI=1TO12:POKE48013+I,ASC(MID‡("ALIEN SCOUTS",I,1)):NEXT
36 PRINTCHR‡(6)CHR‡(17):P‡="P"
40 GOSUB10000
50 IFQQ=0THENGOSUB15000
60 QQ=1
65 GOSUB20000
70 GOTO200
97 REM------------------------------
98 REM              CONTROL
99 REM------------------------------
100 IFI‡=R‡THENX2=X1:X1=X1+1:IFX1=38THENX1=37
101 IFI‡=L‡THENX2=X1:X1=X1-1:IFX1=1THENX1=2
102 IFI‡=F‡THENGOSUB1000
103 IFI‡=P‡THENGETI‡:GOTO106
104 IFI‡<>F‡ANDI‡<>L‡ANDI‡<>R‡ANDI‡<>P‡THENRETURN
105 IFSCRN(X1,20)<>32ORSCRN(X1,21)<>32THEN2000
106 PLOTX1,20,"(":PLOTX1,21,")"
107 PLOTX2,20," "
108 PLOTX2,21," "
109 PLOT10,24,SC‡
110 RETURN
197 REM-----------------------------
198 REM           ALIEN SCOUTS
199 REM-----------------------------
200 CLS:PLOT10,13,14:PLOT10,14,14
201 PLOT13,13,"Alien Scouts"
202 PLOT13,14,"Alien Scouts"
203 WAIT400:EXPLODE:X=19
204 GOSUB5000
206 BA=BA+P
207 IFA1=0THENPLOT10,INT(BA),CHR‡(36):PLOT10,INT(BA-1)," "
208 IFA2=0THENPLOT15,INT(BA),CHR‡(36):PLOT15,INT(BA-1)," "
209 IFA3=0THENPLOT20,INT(BA),CHR‡(36):PLOT20,INT(BA-1)," "
210 IFA4=0THENPLOT25,INT(BA),CHR‡(36):PLOT25,INT(BA-1)," "
211 IFINT(BA)>=22THENGOSUB4000
212 IFA1=1ANDA2=1ANDA3=1ANDA4=1THENGOSUB3000:GOTO300
213 I‡=KEY‡:IFI‡=""THEN206ELSEGOSUB100:GOTO206
297 REM-----------------------------
298 REM           MEGA-BOMBS
299 REM-----------------------------
```

```
300 CLS:PLOT10,13,14:PLOT10,14,14
301 PLOT14,13,"Mega-Bombs"
302 PLOT14,14,"Mega-Bombs":Y=0
303 WAIT400:EXPLODE:X=18:GOSUB5000:I=316
304 ZZ=INT(RND(1)*28+6):Y=Y+1:IFZZ=X1-1ORZZ=X1+1ORZZ=X1+6THENI=315:
GOTO306
305 GOTO320
306 FORZ=8TO22STEPSK/2
307 PLOTZZ,Z,"*":PLOTZZ,Z-SK/2," "
308 MUSIC1,2,3,0:MUSIC2,4,Z/2,0:MUSIC3,3,10,0
309 SOUND4,5,0:PLAY7,0,1,1000
310 IFY>45-SK*2THENGOSUB2000
311 IFE>7+SKTHENGOSUB3000:GOTO400
312 IFZ>21THENSC=SC-15:SC$=STR$(SC):PLOTZZ,Z," ":PLOT10,24,SC$
313 IFZ>21THENCD(ZZ)=1:Y=Y-1:PLOTZZ,Z+1," ":CD(ZZ-1)=1
314 GOTOI
315 WAIT10-SK:NEXT:PLAY0,0,0,0:I=316:GOTO304
316 I$=KEY$:IFI$=""THEN315ELSEGOSUB100:GOTO304
320 ZX=(RND(1)*7)+(SK/3)
321 IFZX<=3.9THEN322ELSE306
322 I$=KEY$:IFI$=""THEN306ELSEGOSUB100:GOTO304
397 REM----------------------------
398 REM            SAUCERS
399 REM----------------------------
400 CLS:PLOT10,13,14:PLOT10,14,14
401 PLOT15,13,"Saucers"
402 PLOT15,14,"Saucers"
403 WAIT400:EXPLODE:X=22
404 GOSUB5000:Y=0
405 Z=INT(RND(1)*13+2):Y=Y+1
406 FORZZ=8TO36STEP1.5+SK
407 PLOTZZ,Z,"&":PLOTZZ-(1.5+SK),Z," "
408 MUSIC3,2,6,0:PLAY2,0,1,1000
409 WAIT10
410 IFY>12-SKTHENGOSUB2000
411 IFE>12+SKTHENGOSUB3000:GOTO7000
412 I$=KEY$:IFI$=""THEN413ELSEGOSUB100
413 NEXT:PLOT34,Z,"    ":GOTO405
997 REM----------------------------
998 REM       FIRING ROUTINE
999 REM----------------------------
1000 ZAP:FORT=20TO1STEP-1
1001 PLOTX1,T,"'":PLOTX1,T+1," "
1002 IFSCRN(X1,T-1)=36ORSCRN(X1,T)=36THEN1010
1003 IFSCRN(X1,T-1)=38THEN1020
1004 IFSCRN(X1,T-1)=42THEN1030
1005 SOUND5,3,0:MUSIC1,2,5,0:PLAY1,2,2,1000
1006 NEXT:PLOTX1,1," ":RETURN
1010 SC=SC+SH*10:SC$=STR$(SC)
1011 IFX1=10THENA1=1
1012 IFX1=15THENA2=1
1013 IFX1=20THENA3=1
1014 IFX1=25THENA4=1
1015 EXPLODE
1016 PLOTX1,T-1," ":PLOTX1,T," ":RETURN
1020 SC=SC+SH*20:SC$=STR$(SC)
1021 EXPLODE:E=E+1:Y=Y-1
1022 PLOTX1,T-1," ":PLOTX1,T," ":RETURN
1030 SC=SC+SH*15:SC$=STR$(SC)
1031 EXPLODE:E=E+1:Y=Y-1
1032 PLOTX1,T-1," ":PLOTX1,T," ":RETURN
1997 REM----------------------------
1998 REM          LIFE LOST
1999 REM----------------------------
2000 CLS:PRINTCHR$(150)SPC(7)"SERIOUS ANNOUNCEMENT":LI=LI-1
2010 PRINT:PRINT"Due to mis-management you have lost a life."
2020 PRINT:PRINTSPC(5)"You have ";LI;" lives left."
2030 IFLI=0THENPLOT14,15,"GAME OVER"ELSE2090
2035 WAIT300
2040 IFSC>HSTHENCLS:PRINTCHR$(149)SPC(10)"New high score"ELSE40
2050 PRINT:INPUT"Name{max.13 characters}=";N$
2060 IFLEN(N$)>13THENPRINT"....naughty":PING:WAIT20:GOTO2040
2070 PLOT5,20,6:PLOT6,20,"Press any key to continue....":GETA$
2075 HS$=SC$
2080 GOTO40
2090 PLOT14,15,"Get ready":WAIT300:BA=4:GOTO200
2997 REM----------------------------
2998 REM        SHEET CLEARED
2999 REM----------------------------
3000 CLS:PLOT10,15,10:PLOT10,16,10
3010 PLOT12,15,"Sheet":PLOT12,16,"Sheet"
3020 PLOT18,15,"Cleared":PLOT18,16,"Cleared"
3030 BO=SH*50+INT(RND(1)*50):SC=SC+BO:SC$=STR$(SC):BO$=STR$(BO)
3040 PLOT13,20,"Bonus :":PLOT20,20,BO$
3050 IFSC>3000ANDFM=0THENLI=LI+1:FM=1
3060 IFSC>15000ANDSM=0THENLI=LI+1:SM=1
3070 A1=0:A2=0:A3=0:A4=0:X1=14
3080 SH=SH+1:E=0:SH$=STR$(SH)
3090 PLOT6,25,"Press any key to continue":GETA$
3100 RETURN
3997 REM----------------------------
3998 REM        ALIENS LANDED
3999 REM----------------------------
4000 CLS
4010 FORY=1TO10
4020 EXPLODE:WAIT15+Y*5
4030 NEXT
4040 PRINTCHR$(150)SPC(7)"SERIOUS ANNOUNCEMENT"
4050 PRINT:PRINT"The aliens have landed,the city is"
4060 PRINT"doomed,flee while you can."
4070 PRINT:PRINT:FORT=1TO10
4080 PRINTSPC(13)"GAME OVER"
4090 NEXT
4100 PRINT:PRINTCHR$(131)SPC(4)"Press any key to continue":EXPLODE:
GETA$
4110 PRINT:PRINTSPC(9)"Your score was ";SC
4115 WAIT300
4120 IFSC>HSTHEN2040
4130 GOTO40
4997 REM----------------------------
4998 REM        SET UP DISPLAY
4999 REM----------------------------
5000 CLS:SHOOT:WAIT100:INK5:CC=28
5010 PLOT1,0,X:PLOT1,23,X:PLOT1,26,X
5~20 FORI=5TO33:IFCD(I)=1THENCC=CC-1:NEXT:GOTO5070
5025 IFCC=0THEN6000
5030 PLOTI,22,"/":NEXT
5070 FORT=1TO4
5080 PLOT1,T,2
5090 NEXT
5100 FORT=5TO8
5110 PLOT1,T,3
5120 NEXT
5130 FORT=9TO13
5140 PLOT1,T,6
5150 NEXT
5160 FORT=14TO19
5170 PLOT1,T,5
5180 NEXT
5190 FORT=20TO22
5200 PLOT1,T,1
5210 NEXT
5220 PLOT3,25,"High score :"
5230 PLOT17,25,HS$
5240 PLOT25,25,N$
5250 PLOT3,24,"Score :"
5260 PLOT17,24,"Sheet :"
5270 PLOT24,24,SH$
5280 PLOTX1,20,"(":PLOTX1,21,")"
5290 PLOT1,24,2
5300 FORT=1TO35
5310 R=INT(RND(1)*36+1):S=INT(RND(1)*16+2)
5320 PLOTR,S,".":NEXT
5330 PLOT10,24,SC$:A=FRE("")
5340 RETURN
5997 REM----------------------------
5998 REM       CITY DESTROYED
5999 REM----------------------------
6000 CLS
6010 FORY=1TO10
6020 EXPLODE:WAIT15+Y*5
6030 NEXT
6040 PRINTCHR$(150)SPC(7)"SERIOUS ANNOUNCEMENT"
6050 PRINT:PRINTSPC(5)"Your city has been destroyed."
6060 GOTO4070
6997 REM----------------------------
6998 REM      CLEARED 3 WAVES
6999 REM----------------------------
7000 CLS:PRINTCHR$(146)
7010 PRINT:PRINTSPC(7)"Well done my son,you have"
7020 PRINT"defeated the alien hordes.Now they"
7030 PRINT"come in greater numbers and strength."
7~40 PRINT:PRINT"To help you,a BONUS has been awarded."
7080 PRINT:PRINT".......an extra base,and,"
7090 PRINT".......500 extra points."
7100 SC=SC+500:SC$=STR$(SC):LI=LI+1:BA=4
7110 P=P+.05:IFP>.6THENP=.6
7120 WAIT750:GOTO200
9997 REM----------------------------
9998 REM        INSTRUCTIONS
9999 REM----------------------------
10000 CLS:PAPER0:INK2:EXPLODE
10010 FORT=1TO6:PRINTCHR$(128+T)SPC(6)"A L I E N  S C O U T S":WAIT
100
10020 EXPLODE:NEXT
10030 PLOT7,15,"Press any key to continue":GETA$
10040 CLS:PRINTCHR$(148)SPC(6)"I N S T R U C T I O N S"
10050 PRINT:PRINT"You control the last missile launcher"
10060 PRINT"of the Inter-Mega-Galactic star fleet"
10070 PRINT:PRINTCHR$(134)SPC(10)"YOUR MISSION"
10080 PRINT"To destroy the Alien Scouts before"
10090 PRINT"they land,and,if this happens,the"
10100 PRINT"game will end.You start with 1 life,"
10110 PRINT"but bonus lives will be awarded for"
10120 PRINT"meritorious performance."
10130 PRINT:PRINTCHR$(134)SPC(10)"YOUR CONTROLS"
10140 PRINT"To move right -";R$:PRINT"To move left  -";L$:PRINT"To
fIre       -";F$
10145 PRINT"To pause      -";P$
10150 PRINT:PRINTCHR$(131)"Are these keys O.K.-":GETA$
10160 IFA$="N"ORA$="n"ORA$="Y"ORA$="y"THEN10165ELSE10150
10165 IFA$="N"ORA$="n"THEN10170ELSE10210
10170 PRINT:PRINT"To move right -":GETR$
10180 PRINT"To move left  -":GETL$
10190 PRINT"To fire       -":GETF$
10195 PRINT"To pause      -":GETP$
10200 IFR$=L$ORR$=F$ORL$=F$ORL$=P$ORR$=P$ORF$=P$THEN10170ELSE10040
10210 PRINT:PRINTCHR$(131)"Select difficulty level{1-5,1=easy}- ":G
ETSK$
10220 SK=VAL(SK$):IFSK<1ORSK>5ORSK<>INT(SK)THEN10210
10230 RETURN
14997 REM----------------------------
14998 REM    RE-DEFINE CHARACTERS
14999 REM----------------------------
15000 FORA1=46368TO46375
15010 READB1:POKEA1,B1:NEXT
15020 DATA30,33,45,33,30,18,18,33
15060 FORA1=46384TO46391
15070 READB1:POKEA1,B1:NEXT
15080 DATA12,18,33,45,45,33,18,12
15090 FORA1=46392TO46399
15100 READB1:POKEA1,B1:NEXT
15110 DATA0,18,18,18,18,0,0,0
15120 FORA1=46400TO46407
15130 READB1:POKEA1,B1:NEXT
15140 DATA0,0,0,0,0,18,18,18
15150 FORA1=46408TO46415
15160 READB1:POKEA1,B1:NEXT
15170 DATA18,30,30,12,63,63,33,33
15180 FORA1=46416TO46423
15190 READB1:POKEA1,B1:NEXT
15200 DATA33,63,12,18,33,18,12,12
15210 FORA1=46424TO46431
15220 READB1:POKEA1,B1:N~XT
15230 DATA63,63,63,63,63,63,63,63
15240 FORA1=46456TO46463
15250 READB1:POKEA1,B1:NEXT
15260 DATA8,10,11,27,31,63,63,63
19997 REM----------------------------
19998 REM       RESET VARIABLES
19999 REM----------------------------
20000 SH=1:X1=14:FORT=5TO33:CD(T)=0:NEXT
20010 SC=0:P=SK*.07+.2:BA=4:LI=1:FM=0:SM=0:A1=0:A2=0:A3=0:A4=0:W=0:
BO=0
20020 SH$=STR$(SH):SC$=STR$(SC)
20030 CLS:RETURN
```

Peter Freebrey

ORIC REVIEW

Colour computing for less than £100? The Oric offers this and more.

The last year has seen the introduction (or proposed introduction) of a number of highly interesting, low cost microcomputers. Sadly most of them have been announced before the final production models have been thoroughly tested and in sufficient quantity to supply the increasing number of would-be purchasers. The Oric also had initial problems and although some writers have suggested that Oric is a derivation of that well known computer aboard the spaceship Liberator its origins actually stem from an anagram of micro (so, they lost an 'm'!).

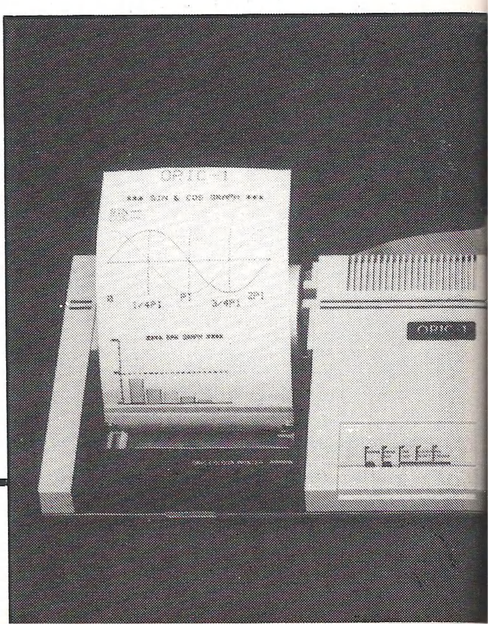
I'm sure Oric Products have done themselves a great disservice in sending out pre-production models for review, as some of the comments I have seen in print are simply not true. Like other micros on the market, the Oric does not conform to any standard of operation and one must spend several hours familiarising oneself with its specific operating system. I stress this as most micros are satisfactory within certain limits but no one computer offers all that the experienced user desires and as certain operations are handled in different ways they require getting used to before useful criticism may be made.

HARD FACTS

The Oric comes complete with power unit, TV lead, demonstration cassette and instruction manual. The early Orics had a very poor 32 page provisional manual that has now been superceded by a much better 164 page ring bound tome and the latest Orics are also being supplied with a cassette lead. The Oric's stated measurements are 280mm by 175mm by 52mm and weighs 1.1kg. It gives the impression of a no-nonsense functional device, the case is made of a heavyweight plastic moulding and is held together with six screws. The keyboard is at a slight angle, being 25mm high at the front and 50mm high at the rear and is clearly labelled and unambiguous. It follows a fairly standard QWERTY layout and the keys have a very positive feel which makes them easy to operate, all keys have auto repeat. The key spacing is similar to a typewriter and touch typing should be perfectly feasible.

General ease of operation is improved by the provision of two Shift keys (at either end of the keyboard), a double size key for Return, a long space bar and individual cursor control keys. On the left hand side are the Escape and Control keys, the Escape key is an enigma as it was only mentioned once in the provisional manual and not at all in the main chapters of the present full manual. Yes . . . if you press ESC followed by various other keys strange thing can happen on the screen but what use can be made of this in 'direct' mode (not within a program) has yet to be explained. Control on the other hand has many uses; eg CTRL 'L' clears the screen; CTRL 'T' toggles from CAPS only to CAPS and lower case; CTRL 'Q' toggles the cursor on and off; CTRL 'C' is a general purpose 'Break' etc, etc. Both Escape and Control are effectively used within programs to perform various functions such as producing double height and flashing characters etc. On pressing any key (except Shift and CTRL) you will get an audible signal (plink!). This is somewhat on the loud side, but not objectionable, and I found it useful when typing in large programs, as there was no need to look at the screen to ensure that what I had typed had actually been accepted (as you have now perceived I am not in the touch typist class). This 'plink' may be switched off by CTRL 'F'.

The case has four soft plastic feet that effectively stop it slipping across the table . . . an important point these days with the advent of such small micros. Sockets and ports at the rear of the case are all unmarked and furthermore both the RGB monitor output and the cassette/audio socket will accept the same plug . . . bad mark Oric. The printer port is for a standard Centronics interface but I have not yet been able to check this. A further expansion port is provided for . . . to quote the manual "extra memory, games cartridges, joysticks and, of course, the modem, this device will allow the Oric to download Prestel pages or even programs and also to send and receive electronic mail via the Prestel Mailbox system". The two remaining sockets are for a standard UHF lead and the



power supply. Accessible from the underside of the case is the fine tuning control for the UHF modulator and the Reset button. The Reset allows you to break out of an infinite loop or from a program crash without losing your program. I have had to use this facility fairly often and wish that it had been sited somewhere other than on the bottom of the case. Normally if you wish to break out of a program or LISTing CTRL 'C' will work well but . . . if you have attempted to CLOAD a program and have not been successful (or change your mind) the only way to regain control is to Reset or take out the power plug and start again.

The Oric uses a 6502 microprocesor and a fairly standard form of Microsoft BASIC which includes such commands as IF . . . THEN . . . ELSE; REPEAT . . . UNTIL; DEEK; DOKE; POP; TRON; TROFF in addition to its variants of colour and high resolution commands.

INNER DEPTHS

The Oric has two principal screen display modes; TEXT and HIRES. In TEXT mode the display is 40 characters wide by 27 characters deep with a black border around the TEXT window. The border colour cannot be changed and the line above the window (within the border) is used for system messages ie Loading, Searching, CAPS etc. Although the screen is 40 characters wide, in practice the first two columns are used as control code positions determining PAPER and INK colours for that row. The character set is standard ASCII and all characters may be re-defined ad these are downloaded into RAM on power up. Characters may be PRINTed or PLOTted to the screen at specified co-ordinates. Control codes may be used within PRINT or PLOT commands so that colour, flashing and double height characters together with cursor control etc, are all accessed within a program. The eight colours available are black, blue, red, magenta, green, cyan, yellow, and white. Control codes may also be POKEd ot PLOTted anywhere on the screen and will affect all characters on the row to their right unless another control code is encountered. In this way you may well have all available colours on the screen at once, the only problem being that with moving characters you must move the control codes too and if you over PRINT (PLOT or POKE) an existing control code then those characters to the right of the new 'PRINT' will assume the new characteristics and lose those originally specified. Control codes appear on the screen as a space. PRINT TAB (N) has a peculiar 'bug' in that TAB (1) to TAB (13) all print to the third column, after this TAB (14) to TAB (50) print ot columns 4 to 40!

In TEXT mode you have two further options; LORES 0 and LORES 1. These offer either the standard character set (LORES 0) or a chunky graphics character set (LORES 1).

In HIRES mode the screen, again with a black border, is divided into two parts the larger upper part is 240 pixels horizontally by 200 pixels vertically and the lower part gives three rows by 40 (38) columns for normal text or commands and is not affected by the upper display.

Control codes (attributes) occupying six pixels horizontally by one pixel vertically may be POKED to the display affecting those pixels to their right giving a similar set of options in colour or flashing as in TEXT mode. The same proviso remains that if a control attribute is overwritten then those pixels on that row to the right of this point lose their original attribute.

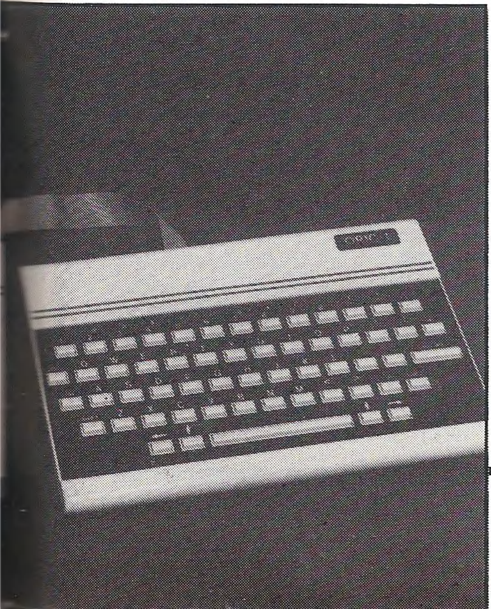
In HIRES you may set the cursor to any point (pixel) on the upper screen, draw CIRCLEs (rather oval!), DRAW lines from one position to another and alter the pattern of these lines (dots/dashes to your choice). You may also FILL A character cells (six pixels by one pixel) by B rows with N value. N may define a defferent colour or pattern or alternatively if you choose the wrong value the screen goes absolutley haywire as you try to FILL with some other control attribute! Normal characters, either the standard ASCII or the chunky graphics set, may be displayed on the HIRES screen by using the command CHAR A,B,C: A being the ACSII code; B which character set and C to specify whether the character should be displayed in foreground or background colour.

Oric's sound capabilities seem to be pretty comprehensive and the internal speaker can give a fair volume of sound! Four pre-programmed effects can be called from BASIC: ZAP; PING; SHOOT and EXPLODE. In addition to these there are three sound commands PLAY, MUSIC and SOUND. There are three sound and one white noise generator. SOUND selects channel, period and volume. MUSIC has been designed to give pure tones and allows selection of channel, octave, note and volume. PLAY controls the envelope shape and can control sounds initiated by SOUND and MUSIC.

An output is provided (on the cassette socket) for connection to your hi-fi. These facilities should give the would-be electronic music composer plenty of scope for experimentation. The only criticism I have of the sound circuitry is that it picks up interference from the rest of the computer and on the internal speaker the ambient buzz and chirr of FOR . . . NEXT loops etc is slightly disquieting (!).

The cassette interface offers

▶



the option of CLOADing or CSAVEing at either 2400 baud or as the manual puts it "super reliable 300 baud". Sadly, although I could CLOAD the demonstration tape at 2400 baud I have been quite unable to CSAVE anything at this baud rate. This could be the fault of my tape recorders (I tried three!) as cassette recorders do vary somewhat. . .but, anyway, "super reliable 300 baud" works every time and I am a great one for reliability. The present machines supposedly allow you to SAVE entire blocks of memory. The manual says that the machine can SAVE and LOAD blocks of memory between specified locations. It also implies that this option may be used within a BASIC program; sadly this is not possible at the moment as this operation interferes with the operation of a BASIC program. In my opinion one serious omission from the cassette routines is the lack of any sort of VERIFY command. I'm sure I'm not alone in feeling very edgy not knowing whether that valuable program is really saved on tape for prosperity. . .or not! I believe that Oric Products are working on routines (ROM mods?) to enable the saving of arrays, let's hope they also have VERIFY in their minds too! (Oric have assured us that they do. Ed.)

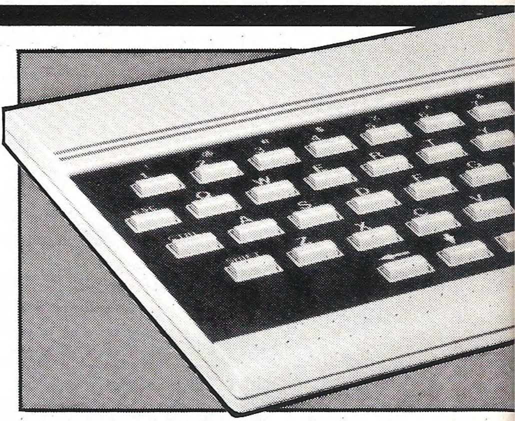
	ORIC	SPECTRUM	ORIC (with keyboard inhibited)
BM1	2.1	4.9	1.6
BM2	17.8	9.0	14.0
BM3	29.7	21.9	23.3
BM4	32.1	20.7	23.3
BM5	38.7	25.2	25.1
BM6	52.7	68.2	30.4
BM7	78.8	86.7	41.3
BM8	23.6	25.1	61.6
			18.5

Table 1. Results of Benchmark tests.

DOCUMENTATION

Some mention of the instruction manual must be made. as with any new piece of equipment one should always study this first! The full manual is now available and is not bad as manuals go but it suffers a number of errors and omissions. There is an errata sheet that should go with it. . .make sure you get yours. There is nothing quite so disheartneing as typing in your first program and for it not to work. The errors in program examples are for the most part fairly obvious and provide excellent practice at debugging but even so. . .(Oric tell us that they have in fact commissioned a totally new manual which should be avialable in Autumn 1984. The existing manual has been reprinted to correct all errors and an index has now been also added Ed).

The manual covers most things a beginner would need to know: setting up the computer; BASIC programming; how to get colour onto the screen; how to edit your programs; number crunching; how to handle strings; what to do with Oric's sound and how to save your programs on tape. There is even a chapter on machine code programs (it doesn't tell you much but does suggest that it might be worthwhile learning a bit more). There is also a chapter on using you printer but chapter 15 is probably the most vital to everyone, as this lists Oric's BASIC commands and is going to be looked at before anything else, certainly by those who have used computers before. The manual ends with 11 appendices that cover a variety of useful items. One very useful item supplied with your Oric is the demonstration tape! It is a bit slow and boring to watch in action — but is quite a useful reference when you LIST the program and see how various operations were programmed to happen. For instance when you



try to double height character program in the manual and it doesn't work. . .see how it is done on the demo tape!

IN USE

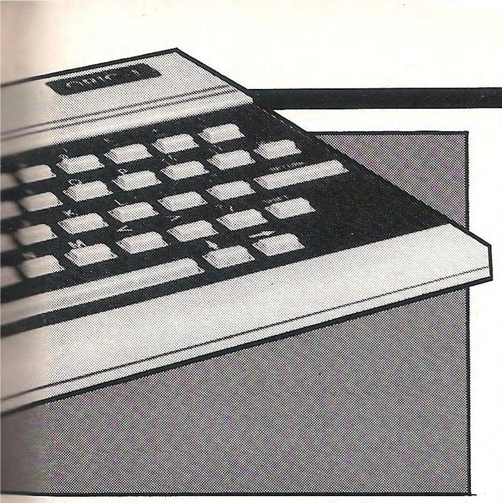
In operation the Oric is an easy machine to use and I found that after some hours at the keyboard there was no indication of fatigue or strain. The error messages are in plain English and seem to amke sense. Line numbers up to 63999 are accepted, you need only DIMension an array if it is to have over 10 elements. Variable names may be a single letter or a letter followed by a single integer, numeric variables may be defined as integer variables by the suffix '%'.

Like many other computers, some arithmetic routines can lead to some surprising results: I ran a small routine within a FOR. . .NEXT loop adding an increment of 0.2 on each pass. I then PRINTed the total and the INTeger. On my total reading 5, I was informed by the computer that the INT (of this number) was 4! Oh well, I never did trust computers to work out my wages. . .

String arrays may have up to 255 characters and do not need the length of the string to be dimensioned.

It is not very fast in operation, the standard set of 'benchmarks' were used to measure the relatvie speed of operation and the results are shown in Table 1.

The Oric can be speeded up slightly if you do not need to use the keyboard during a given routine; type in line no. CALL E6CA before any routine processing of data and don't forget to type in line no. CALL E804



after the routine and before you need to access the keyboard. This inhibits the scanning of the keyboard for any entry so if you forget the second CALL you will not be able to use the keyboard even when the program has finished RUNning! The Benchmark results were similar to the Sinclair Spectrum so I have included these for comparison.

One early program with the Oric was to discover how to EDIT programs. The provisional manual was not very explicit on the matter at all. It required you to use CTRL 'A' and ESC to inser characters in a program line. The full manual puts over editing in a much clearer manner, explaining how you may alter characters within a program line using CTRL 'A' to copy a line up to the offending character and then typing in the correct character and continuing to copy the rest of the line with CTRL 'A'. So far so good, but what if you want to insert a character or even several characters? It's easy when you know how (isn't everything), again copy over the line using CTRL 'A' until you get to the position where you want to insert additional characters. Now use the cursor control keys to move away from the program line (up or down) type in your insertion and use the cursor control keys to return to where you left the program line. Continue copying the line using CTRL 'A' to the end of the line, press Return and the job's done. Whew. . .sounds complicated but you very soon get used to it. All this may be done directly over a LISTing or after having typed EDIT (line).

IN CONCLUSION

One cannot escape the fact that in size, cost and capabilities the Oric is bound to be compared with the Sinclair ZX Spectrum. A lot of people are going to be asking 'why should I buy one'?

As is often the case when comparing microcomputers of similar cost, it is difficult to say outright that one is better than another. This has nothing to do with trying to be nice to everyone, it is more a case of one man's computer is another man's millstone and vice versa!

The Oric has a slightly more versatile BASIC than the ZX Spectrum; IF. . .THEN. . .ELSE, REPEAT. . .UNTIL and TRON for example. On the other hand, the ZX Spectrum will not permit you to Enter a program line that has a syntax error. The ZX Spectrum has such commands as VERIFY and MERGE (to append one program in memory with another from tape).

Both Oric and ZX Spectrum have a respectable amount of memory available to the user.

I think in both cases I would recommend the 48K versions if you can possobly afford them.

Moving graphics are almost certainly going to be easier to implement on the ZX Spectrum. The ZX Spectrum does not have a FILL command from BASIC on the other hand it isn't terribly difficult to find a routine to fulfil a similar funtion. HIRES colour resolution is six times better on the Oric but don't forget thatyou have to make allowance for those control codes (six pixels long). The Oric looks as though it will be more readily versatile in its accessibility to the outside world: printers, monitors modems etc. So far I have not been able to find the 80 characters that the Oric was supposed to have allocated for user defined characters. . .the ZX Spectrum definitley has 21.

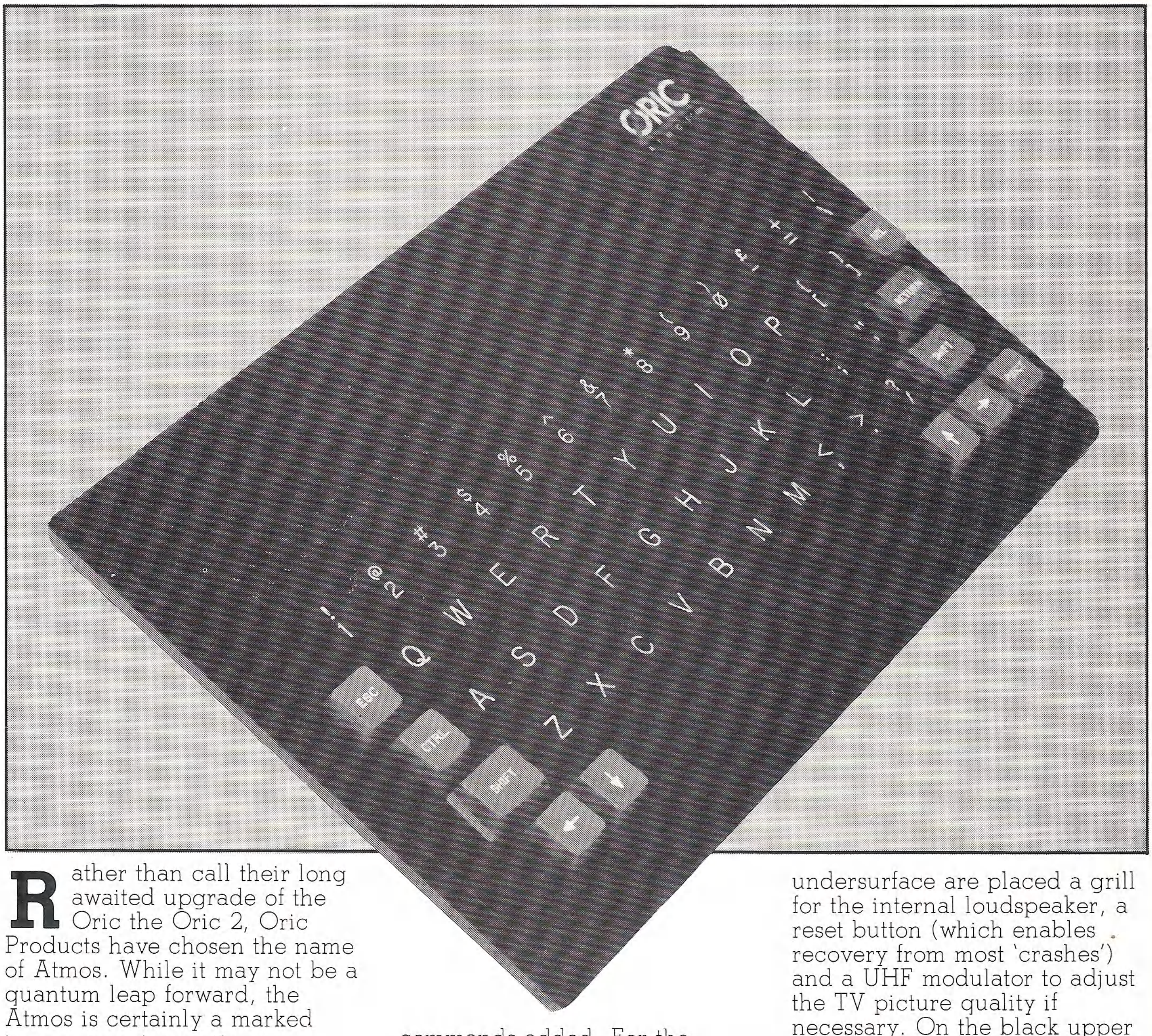
The screen format of the Oric (40 by 27) is more normal than the ZX Spectrum's (32 by 22) but the ZX Spectrum's manual is definitley superior to that of the Oric. The Oric has better sound capabilities but the ZX Spectrum has a large amount of software being produced for it (I have no doubt that a number of software houses are already far advanced on software for the Oric). As you can see nearly all of the comparisons seem to be 'slightly better than' and in an overall decision it is very difficult not to let ones own likes and dislikes cloud the impartial view. I think that the beginner will find the keyword system of te ZX Spectrum very easy to pick up, but in the long term, the Oric could prove to be more versatile. I think that the Oric wil certainly be up there with the ZX Spectrum on a great many peoples short lists.

NOTE: This review appeared in its original form in *Computing Today,* May 1983. Efforts have been made to ensure that this is accurate and current.

FACTSHEET	Oric 1		
CPU	6502A		
ROM	16K	Supplier	Oric Products
RAM	16K		International Ltd
Language	BASIC		Coworth Park
Keyboard	QWERTY		London Road
	Cursor controls		Ascot
Display	Text mode: 28 by 40		Berkshire SL5 7SE
	Graphics mode: 200 by 240		0990-27641
	Eight colours		
	80 user definable characters		
Cassette	300 or 2400 baud		
I/O	Parallel port		
	Printer port		
	Expansion bus		
Sound	Three channels		
Options	64K RAM with 16K overlaid ROM		
	Printer		
	Disc drive		
Costs	Oric 1 16K RAM		£99.95
	Oric 1 48K RAM		£139.95
	Oric MCP-40 printer/plotter		£169.95 (£40 voucher supplied with every Oric)
	Hitachi 3" disc drive		under £200 (available in Autumn 1983)

ATMOS REVIEW

Oric have given their Micro a facelift. But are the changes purely cosmetic?



Rather than call their long awaited upgrade of the Oric the Oric 2, Oric Products have chosen the name of Atmos. While it may not be a quantum leap forward, the Atmos is certainly a marked improvement over its predecessor.

BASIC CHANGES

Oric 1 owners will notice three basic changes. The livery of the casing is now red and black as opposed to two tone grey. The computer has a 'real' keyboard, and a new 16K ROM is fitted, with the well known 'bugs' removed and a few additional commands added. For the benefit of those readers who are not too familiar with the Oric 1, let's take an in depth look at the Atmos.

DIMENSIONS AND DESIGN

The computer measures 11″ by 7″ and stands 2″ high at the back and 1″ at the front, providing a comfortable slope to the keyboard. On the red undersurface are placed a grill for the internal loudspeaker, a reset button (which enables recovery from most 'crashes') and a UHF modulator to adjust the TV picture quality if necessary. On the black upper surface is the 58 key typewriter style keyboard. The keyboard is one of the best features of the Atmos and I believe, is comparable with the best in current home computers. It has a comfortable 'feel' and any typist should take to it with ease. The standard typewriter keys are black in colour with clear white lettering. The computer 'control' keys are coloured red — a nice idea which helps the computer novice.

The Atmos has a misleading FUNCTion key. At a glance this might suggest that the Atmos has user defined single key functions. It does not. I'm told that the same keyboard will be used in future Oric computers, which will have single key functions. On the Atmos, the FUNCT key has no function!

The Atmos is connected to the outside world via the clearly marked ports at the back. DIN sockets provide input/output for the cassette back-up store (and hi-fi sound output), and output to an RGB monitor. Alternatively, the screen output can be relayed to a standard black and white or colour TV via the UHF output. In addition to the power input, the remaining sockets at the back are a general expansion port and a printer port, which connects to the Oric printer, or to any other printer with a Centronics interface.

DOCUMENTATION

A DIN to DIN cassette lead, TV lead, power back, a demonstration tape, and a manual are supplied with the Atmos. The tape provides a very good demonstration of the excellent sound and graphics capabilities of the Atmos (plus an indifferent game). Oric have taken a lot of care with the Atmos manual. They have provided a well produced 194 page paperback book, which caters for the beginner whilst supplying at least some of the information normally required by the more expert programmer.

TECHNICAL DETAILS

The Atmos is driven by the well-tried and trusted 6502 processor. A standard Microsoft BASIC (expanded to include facilities for sound, high resolution graphics and colour) is available on a 16K ROM. The operating speed is about the same as the Oric 1 (which, in turn, is about the same as the Spectrum), which makes it slow compared to most other micros.

Two display modes are provided, TEXT and HIRES. In text you have access to 27 rows of 49 characters length. There are two character sets, the usual alphabetic, numeric, and punctuation mark set, and a set similar to teletext graphics. Both character sets can appear on the screen at the same time and any character can be redefined. Characters can be printed normally, or as inverse, double height, and flashing.

In HIRES mode, the screen is divided into two parts. The majority is a bit-mapped 240 by 200 pixel display for high resolution graphics. At the bottom of the screen are three lines of normal TEXT. High resolution drawing uses a cursor, which is positioned using CURSET and moved with CURMOV. The DRAW command gives you straight lines, and the CIRCLE command — well that's self explanatory. You can produce dotted lines and circles with the PATTERN command. Other useful high resolution commands are FILL, which allows you to fill a specified area with a pattern or colour, and CHAR, which prints characters on the HIRES portion of the screen.

In both display modes, you can have eight foreground and eight background colours. INK and PAPER commands control the overall colours. For more colours on the screen at once, attributes have to be 'printed' onto the display file. This system of serial attributes can be difficult for the beginner to master, but, as the software for the Oric 1 demonstrates, excellent colour and resolution are possible using the system.

The Atmos uses a 8912 sound generator chip, which can produce really impressive sound effects. Four pre-programmed sounds, ZAP, PING, SHOOT, and EXPLODE are available, and you can use SOUND, MUSIC and PLAY to create your own sound effects.

RELIABLE LOADING AND SAVING

Cassette loading and saving proved to be reliable. The Atmos has two baud rates 2499 and 3000 and I observed no problems with either. The Atmos allows program verification, and allows programs to be joined inside the computer. Two new commands STORE and RECALL allow arrays to be saved and recalled to and from tape. Searching, found and loading messages provide a clear guide of the status during program and data loading. The only annoying feature of cassette use is the incorrect 'Load error' message which sometimes appears after loading a program but the handbook explains how this can be overcome.

CONCLUSIONS

The Atmos is currently available with 48K of RAM, and is priced at £170. A 16K version should follow. It may appear a little old-fashioned alongside some of the other 'new' computers of 1984, and the price a shade high, but I believe that Oric Products have at last realised their boast of last year's advertisements — they have a quality computer. Oric 1 owners who agree will have the opportunity of upgrading to the Atmos. But if you already have a lot of Oric 1 software, treat claims that these programs will work on the Atmos with caution. I tried several and a few did not work. As a general rule, programs written entirely in BASIC should work (although INVERSE and NORMAL have been replaced by STORE and RECALL, and PLOT is slightly different). Recent machine code offerings for Oric-approved software houses should also be OK (the new ROM has been around for several months), but older machine code programs may crash, if they depend upon ROM facilities.

ATMOS SYSTEM	Price
48K Processor (37K usable memory) inc. leads and adaptor	£170
16K Expansion Pack (not available yet)	
160K Micro Disc Drives	
4 colour Printer/plotter (ball point pen)	£149.95

Simon Lucas

ESCAPE FROM DANGER ISLAND

All you need is £8 to get you off Danger Island and safely home; we explain how to earn your fare.

It was bad enough choosing Danger Island for your holiday, but now you've run out of money how do you ever get home?

Well, there is one way. Scattered around the island are eight valuable objects. If you can collect them and take them to the pawnbrokers, you can get £1 each for them. Collect all eight and you have just got enough money for your fare home.

The treasures you must find are: a teapot, horseshoe, gold nugget, Chinese kite, Swiss Army knife, silver needle and a wedding ring.

The program will respond to two-word phrases. The words you can use are:

NSEW
INVENTORY
SCORE
GRAB
LEAVE
SWIM
JUMP DOWN
GO DOWN
GO OUT
LOOK
EAT
SEARCH
EXAMINE
GIVE
GET
TAKE
FET
DROP
PUT
CLIMB
OUT
IN
GO IN
THROW
DRINK
HELP
DIG

The program was written in a 48K Oric, but should work on the 16K model.

Before you type the program in, there are two points you should note. The listing was produced on a printer which prints the £ sign as __ (an underscore), so you should change this back.

Second, you may find some lines seem to be too long to enter. They will fit if you enter PRINT as the shortahnd ? (question mark).

One problem with listing Oric programs is that the printer will randomly produce squiggles. This is caused by the keyboard scanning interrupts interferring with the output from the printer port.

XS(x)	hold messages
P%	holds the number of the current location
S%(x,y)	holds the number of the location you visit if you go N,S,E,W
V$(x)	holds what you are carrying
Q$(x)	holds a description of the location
G$(x)	holds items found at each location
N$(x)	holds words recognised (second word entered)
B%(x,1),	act as pointers to other arrays
N%(x)	test flag
R	the number of the word entered
Z$	your response to what should I do. This is split in two parts **C$** first word and **L$** is the second word
H,D	act as counters for loops
A%	score

Other variables act as flags to test if you have done certain things in the game, such as dropped the food.

HOW IT WORKS

40	selects TEXT mode & colour of foreground/background
45	selects subroutine for titles etc.
50	DIMensions the arrays
60	sends it to two subroutines to check the score and to read data into the arrays
90-155	display directions you can go in
101	acts as garbage collection routine (may be omitted from oter machines)
160-185	display messages if you have solved the game
190-240	display what you can see and where you can go
250	waits for your response
270-460	recognise your words and send to appropriate routine
3000-3020	titles
10000-110220	DATA for locations
10230-10290	DATA for items you find
10300-10330	DATA for words understood
11000-11090	score routine
12000-12050	reads DATA into arrays
13000-13480	routine for GET responses
13500-13570	splits input response into two words
14000-14500	drop routine
14700-14740	inventory routine
15000-15020	swearing not allowed routine
17000-17040	swim routine
17500-17540	death routine
17600-17800	throw routine
18000-18015	search routine
19000-19010	dig routine
19500	give routine

To produce a squiggle-free listing, all you need to enter is:
CALL (hash sign)
E6CA:LLIST:CALL (hash sign)
E804

HINTS ON CONVERSION

The program is in a fairly standard version of BASIC and will require only minor modification to run on most machines.

Line 40 may be deleted from most machines, as it only selects the text mode and colours used. On the BBC micro use MODE 7.

Line 101 is a garbage collection routine and may be deleted from most machines entirely.

The CHR$ codes from 129-135 are used to change colour. They may be changed to suit your machine. **Note:** on the BBC micro, the same codes are used.

The PLOT commands, 3002-3020, print the text at different locations on the screen. The numbers are X co-ordinate, Y co-ordinate, text or X,Y, colour code (0-7). This may be replaced with PRINT TAB(X,Y) on BBC, PRINT @, on TRS80 etc.

Integer variables have been widely used as they are more economical of memory space. They may be changed to ordinary variables if required.

```
10 REM DANGER ISLAND
20 REM A MINI-ADVENTURE FOR THE ORIC1 (16K & 48K)
30 REM BY S.W. LUCAS
40 TEXT:GRAB:PAPER0:INK1
45 GOSUB3000
50 DIMQ$(50),G$(50),S%(50,4),V$(4),B%(60,1),N$(60),X$(30),N%(60)
55 X$(30)="Even Tarzan couldn't carry any more!!!and I'm not that s
t~ong!!"
60 A$="":P%=1:RESTORE:GOSUB11000:GOSUB12000
90 CLS
100 PING:PRINTCHR$(131)"I am ":PRINTQ$(P%):A$=""
101 A=FRE(""):REM GARBAGE COLLECTION ROUTINE
110 IFS%(P%,1)>0THENA$="NORTH"
120 GOSUB11000
130 IFS%(P%,2)>0ANDLEN(A$)>0THENA$=A$+",SOUTH"ELSEIFS%(P%,2)>0THENA
$="SOUTH"
140 IFS%(P%,3)>0ANDLEN(A$)>0THENA$=A$+",EAST"ELSEIFS%(P%,3)>0THENA$
="EAST"
150 IFS%(P%,4)>0ANDLEN(A$)>0THENA$=A$+",WEST"ELSEIFS%(P%,4)>0THENA$
="WEST"
155 IFA$=""THENA$="Nowhere at all"
160 IFA%=8THENCLS:PAPER7:INK1:PRINT:PRINT:PRINT:PRINT:PRINT"WELL DO
NE YOU HAVE
170 IFA%=8THENPRINT:PRINT:PRINT"FOUND ALL 8 ITEMS OF TREASURE AND"
180 IFA%=8THENPRINT:PRINT:PRINT"EXCHANGED THEM FOR YOUR FARE HOME !
!"
185 IFA%=8THENFORZ=1TO10:ZAP:WAIT20:NEXT:END
190 PRINT:PRINTCHR$(129);"I can go :- "A$
200 E=0:FORT=1TO19:PP%=0:IFB%(T,1)=P%THENPP%=1
210 IFPP%=1THEN230
220 NEXT:GOTO250
230 IFE=0ANDG$(T)<>""THENPRINT:PRINTCHR$(133);"I can see :-"
240 PRINTG$(T):E=E+1:GOTO220
250 PRINT:PRINTCHR$(131)"What do I do ";:INPUTZ$
260 B$=LEFT$(Z$,2):C$=LEFT$(Z$,3)
270 CLS:PRINTCHR$(133)"HOLD ON A SEC!"
271 IFZ$="GO IN"ORB$="IN"THEN IFP%=20 THENP%=21:GOTO100
272 IFZ$="GO OUT"ORB$="OU"THEN IFP%=21 THENP%=20:GOTO100
275 IFB$="GO"THENPRINT"use N,S,E or W":GOTO100
280 IFB$="SC"THENGOSUB11000:PRINTCHR$(131);"you have made _";A%;" a
nd need _8"
290 IFB$="N"ANDS%(P%,1)<>0THENP%=S%(P%,1):GOTO100
300 IFB$="S"ANDS%(P%,2)<>0THENP%=S%(P%,2):GOTO100
310 IFB$="E"ANDS%(P%,3)<>0ANDP%<>17THENP%=S%(P%,3):GOTO100
311 IFB$="E"ANDP%=17ANDAB<>1THENPRINT"The dog won't let me go anywh
ere":GOTO100
312 IFB$="E"AN~P%=17THENP%=S%(P%,3):GOTO100
320 IFB$="W"ANDS%(P%,4)<>0THENP%=S%(P%,4):GOTO100
325 IFB$="FU"ORB$="PI"ORC$="BUG"THENGOSUB15000
330 IFB$="GE"ORB$="TA"ORB$="GR"THEN13000
340 IFC$="DRO"ORB$="LE"ORB$="PU"THEN14000
350 IFC$="INV"THEN14700
360 IFC$="SWI"THEN17000
```

```
365 IFC$="CLI"THENIFP%=10THENP%=22:GOTO100ELSEIFC$="CLI"THENPRINT"H
OW":GOTO100
366 IFB$="UP"THENIFP%=10THENP%=22:GOTO100
370 IFC$="JUM"ANDP%=22THEN17500
372 IFC$="JUM"THENPRINT"O.K. I don't get very far !!":GOTO100
375 IFC$="DW"ANDP%=22THENP%=10:GOTO100
376 IFC$="DOW"THENPRINT"DON'T BE STUPID":GOTO100
380 IFC$="LOO"THEN100
385 IFC$="THR"THEN17600
390 IFC$="EAT"THENPRINT"I'm not very hungry thank you!":GOTO100
395 IFC$="DRI"THENPRINT"There's nothing here to drink !":GOTO100
400 IFC$="HEL"THENPRINTCHR$(132);"Im not going to solve it for you
!":GOTO100
401 IFC$="EXA"THENPRINT"I can't see anything special !":GOTO100
405 IFC$="SEA"THEN18000
410 IFC$="DIG"ANDP%=23THEN19000ELSEIFC$="DIG"THENPRINT"not here!":G
OTO100
415 IFC$="GIV"THEN19500
450 PRINT"sorry!":GOTO100
460 END
2999 END
3000 CLS:PRINTCHR$(17)
3001 FORX=1TO2
3002 PLOT9,X,X:PLOT10,X,14:PLOT11,X,"DANGER ISLAD" : NEXT
3003 PLOT1,10,5
3004 PLOT2,10,"You are a student fallen on hard"
3005 PLOT1,11,5
3006 PLOT2,11,"times. You have gone on a long needed"
3007 PLOT1,12,5
3008 PLOT2,12,"holiday but have run out of money."
3009 PLOT1,20,3
3010 PLOT2,20,"Your task is to find sufficient items"
3011 PLOT1,21,3
3012 PLOT2,21,"of value and take them to the "
3013 PLOT1,22,3
3014 PLOT2,22,"pawnbrokers to exchange for cash ."
3015 PLOT1.25.1
3016 PLOT2,25,"You need 8 pounds for your fare home"
3017 PLOT2,26,"and each treasure is worth _1"
3020 RETURN
9999 END
10000 DATAon a footpath with trees on either    side.,0,0,2,0
10010 DATAon the shores of a large lake,0,3,0,1
10020 DATAon a mountain track,2,4,0,0
10030 DATAat the top of a mountain. I can see   for miles.,3,0,5,6
1~040 DATAby a waterfall,0,10,0,4
10050 DATAon a narrow mountain track. It is very misty here.,0,7,4,
0
10060 DATAby a rock fall. There are boulders     everywhere.,6,8,0,
0
10070 DATAby a cave entrance;7,0,9,0
10080 DATAin a vast cavern with stalagtites      everywhere.,0,0,11
,8
```

```
10090 DATAby the waterfall. There is a very tall tree here.,5,11,0,
0
10100 DATAin the waterfall. I can see a cave   'entrance.,10,0,12,9
10110 DATAat the other side of the waterfall.,0,13,0,11
10120 DATAon a footpath,12,14,0,0
10130 DATAat the bottom of the mountain.There isa small village.,13
,16,15,17
10140 DATAon a main road.,18,0,0,14
10150 DATAby the village blacksmiths. It is     closed.',14,0,0,0
10160 DATAby some houses.There is an old lady   here.,0,0,14,0
10170 DATAon a road bridge.,0,15,19,0
10180 DATAon the main road,20,0,0,18
10190 DATABy a pawnbroker's shop,0,~9,0,0
10200 DATAinside the pawnbroker's shop. Drop    your treasures here
!,0,0,0,0
10210 DATAup a tree,0,0,0,0
10220 DATAon a small island,0,0,0,0
10230 DATAan old empty beer can,12,some algae,11,a small boy,4
10240 DATA"",7,a tiny £GOLD£ nugget,9,a £HORSESHOE£,16,a £TEAPOT£,1
7
10250 DATA"",23,a £WEDDING RING£,23
10260 DATAa valuable Chinese £KITE£,22
10280 DATApebbles,11,stones,7,a squirrel,13,nuts,1,a shovel,23
10290 DATAa hammer,16,a dog,17,some dog food,1,a large boulder,8
10300 DATABEER,1,CAN,1,ALGAE,2,BOY,3,SWISS,3,BADGE,4,ROLLS,4,GLD,5,
NUGGET,5
10310 DATAHORSESHOE,6,SHOE,6,TEAPOT,7,SILVER,8,NEEDLE,8,WEDDING,9,R
ING,9
10320 DATACHINESE,10,KITE,10,PEBBLES,11,STONES,12,SQUIRREL,13,NUTS,
14,SHOVEL
10330 DATA15,HAMMER,16,DG,17,FOOD,18,BOULDER,19
11000 A%=0
11030 IFB%(3,1)=21 THENA%=A%+1
11040 IFB%(4,1)=21 THENA%=A%+1
11050 IFB%(5,1)=21 THENA%=A%+1
11060 IFB%(6,1)=21 THENA%=A%+1
11070 IFB%(7,1)=21 THENA%=A%+1
11072 IFB%(8,1)=21 THENA%=A%+1
11075 IFB%(9,1)=21 THENA%=A%+1
11080 IFB%(10,1)=21THENA%=A%+1
11090 RETURN
12000 FORH=1TO23:READQ$(H):FORD=1TO4:READS%(H,D):NEXTD,H
12010 FORH=1TO19:READG$(H),B%(H,1):NEXT
12040 FORH=1TO27:READN$(H),N%(H):NEXT
12050 RETURN
12999 REM GET ROUTINE
13000 GOSUB13500:IFL%=1THEN13020
13010 GOTO100
13020 E%=0:FORH=1T~19:IFB%(H,1)=P% ANDB%(N%(R),1)=P%THENE%=1
13030 NEXT
13040 IFE%=0THEN100
13041 IFR=23THENAA=AA+1
13042 IF(R=4ORR=5)ANDAC<>1THENPRINT"I  can't do that dummy !":GOTO1
00
13043 IFR=25THENPRINT"It has savagely attacked me aaaggghh!":GOTO17
510
13450 E%=0:FORD=1TO3:IFV$(D)=""THENV$(D)=G$(N%(R)):E%=1:D=5
13460 NEXT
13342 IFE%=0THENPRINTX$(30):GOTO100
13480 B%(N%(R),1)=0:GOTO100
13499 REM ROUTINE TO MAKE SENSE OF RESPONSE
13500 L$="":FORH=1TOLEN(Z$)
13510 IFMID$(Z$,H,1)=" "THENL$=RIGHT$(Z$,(LEN(Z$)-H)):H=H+40
13520 NEXT
13530 R=0
13540 L%=0:IFLEN(L$)<2THENRETURN
13550 FORH=1TO27:IFLEFT$(N$(H),LEN(L$))=L$ THENL%=1:R=H
13560 NEXT
13570 RETURN
13999 REM DROP ROUTINE
14000 GOSUB13500
14010 IFL%=1THEN14030
14020 PRINTCHR$(131)"I can't see a ";L$:GOTO100
14030 E%=0
14040 FORD=1TO3:IFV$(D)=G$(N%(R))THENV$(D)="":E%=1
14050 NEXT
14060 IFE%=1THEN14080
14070 PRINTCHR$(130);"I've not got it dummy!!!!":GOTO100
14080 B%(N%(R),1)=P%:CLS
14090 IFR=23THENAA=0
14100 IFP%=17ANDR=26THENAB=1:PRINT"It is busy eating. I think I can
      escape"
14500 GOTO100
14699 REM INVENTORY ROUTINE
14700 CLS:PRINTCHR$(130);"I have :- "
14705 F%=0
14710 FORH=1TO3:IFV$(H)<>""THENPRINTCHR$(131);V$(H):F%=1
14720 NEXT
14730 IFF%=0THENPRINT"Nothing at all"
14735 PRINT
14740 GOTO100
15000 CLS:PRINT:PRINT:PRINT:PRINTCHR$(131)"How dare you talk to me
like that?"
15010 PRINT:PRINT:PRINTCHR$(132)"What have you got to say to me now
?"
15020 REPEAT:INPUTZ$:ZAP:UNTIL Z$="SORRY":RETURN
16999 REM SWIM ROUTINE
17000 IFP%=2ORP%=23THEN17020
17010 PRINT"I can't swim here! IDIOT!":GOTO100
17020 PRINT"O.K."
17030 IFP%=2THENP%=23ELSEIFP%=23THENP%=2
17040 GOTO100
17499 REM DEATH ROUTINE
17500 CLS:PAPER7:INK1:PRINT:PRINT:PRINT:PRINT"I FELL AND BROKE MY N
ECK "
17510 PRINT:PRINT:PRINT"I AM DEAD"
17520 PRINT:PRINT:PRINT"Do you want to play again?"
17530 GETA$
17540 IFA$="N"THENEND ELSE RUN
17599 REM THROW ROUTINE
17600 GOSUB13500
17610 IFL%=1THEN17630
17620 PRINT:PRINT"I can't see a ";L$:GOTO100
17630 E%=0
17640 FORD=1TO3:IFV$(D)=G$(N%(R))THENV$(D)="":E%=1
17650 NEXT
17660 IFE%=1THEN17680
17670 PRINT"I've not got it !!! dummy!!":GOTO100
17680 IFP%<23THENB%(N%(R),1)=P%+1ELSEB%(N%(R),1)=P%
17690 PRINT"O.K"
17800 GOTO100
17999 REM SEARCH ROUTINE
18000 IFP%=7THEN18010ELSEPRINTCHR$(130)"I can't find anything":GOTO
100
18010 G$(4)="A £BADGE£ from a ROLLS ROYCE car"
18015 IFAZ<>0THENPRINT"There's nothing else here!":GOTO100
18020 PRINT"I have found something !!!":AZ=1:GOTO100
19000 IFAA<>1THENPRINT"I don't have anything to dig with":GOTO100
19010 PRINT"I have found something":G$(8)="A £SILVER £ needle":AA=2
:GOTO100
19500 GOSUB13500:IFL%=1THEN19515
19510 GOTO100
19515 IFP%=4ANDR=22THENPRINT"He takes a few and offers me something
"
19520 IFP%=4ANDR=22THENPRINT"in exchange. He then runs off singing"
19530 IFP%=4ANDR=22THENG$(3)="A £SWISS ARMY KNIFE£":N$(4)="KNIFE":A
C=1:GOTO100
19540 IFP%=21ANDR=22THENPRINT"The squirrel doesn't like them!":GOTO
100
19550 PRINT"I'm not quite sure I know what you    mean":GOTO100
```

ANIROG

NEW TURBO 64 GAMES

P.C. FUZZ

Aggro at closing time at the local. Getaway car screeches to a halt outside a bank. Ingenious methods employed by the Mafia to literally spirit away the loot from the High Street. Never fear - P.C. Fuzz is on patrol.

COMMODORE 64 £7.95

ZARGA MISSION

An action packed 3D scrolling arcade thriller by the author of Space Pilot.

COMMODORE 64 £7.95

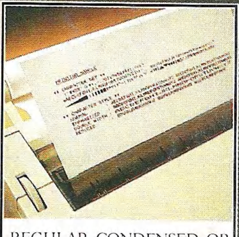
Also available on Disk at £9.95

TRADE ENQUIRIES: ANIROG SOFTWARE LTD. 29 WEST HILL DARTFORD KENT (0322) 92513/8
MAIL ORDER: 8 HIGH STREET HORLEY SURREY 24 HOUR CREDIT CARD SALES HORLEY (02934) 6083
PAYMENT BY CHEQUE P.O. ACCESS/VISA 50p POSTAGE & PACKAGING

Little Brothers should be seen but not heard.

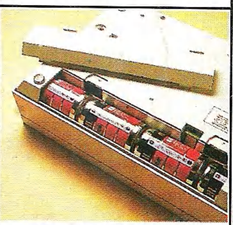
brother
THERMAL TRANSFER PRINTER
HR-5
brother HR-5
LINE FEED
ON LINE
ALARM
PAPER EMPTY
RIBBON END

NOW ONLY
£159.95
inc.VAT.



REGULAR, CONDENSED, OR EXTENDED FACES.

CUT SHEET A4 OR ROLL PAPER.



BATTERY OR MAINS OPERATED.

A maxim which eloquently describes the Brother HR-5.

Less than a foot across, it's nonetheless loaded with features.

The little printer that's low on decibels.

There's one thing the HR-5 won't give you.

Earache.

For the annoying 'clickety clack' many printers produce is mercifully absent from the HR-5.

Quietly efficient, it delivers high definition dot matrix text over 80 columns at 30 characters per second (maximum).

Text or graphics with ease.

The HR-5 also has something of an artistic bent.

Being capable of producing uni-directional graphics and chart images together with bi-directional text. What's more it will hone down characters into a condensed face, or extend them for added emphasis.

At home with home computers.

Incorporating either a Centronics parallel or RS-232C interface, the HR-5 is compatible with BBC, Spectrum, Oric, Dragon, Atari and most other home computers and popular software.

Perfectly portable, the battery or mains operated HR-5 weighs less than 4 lbs, and has a starting price of only £159.95 (inc. VAT).

Which is really something to shout about.

PLEASE SEND ME MORE DETAILS OF THE REMARKABLE BROTHER HR-5 PRINTER.

NAME____________________

ADDRESS____________________

PCT__________ TEL NO.__________

AVAILABLE FROM: BOOTS, RYMANS, WILDINGS, SELFRIDGES AND ALL GOOD COMPUTER EQUIPMENT STOCKISTS.

brother®

DEPT P, BROTHER OFFICE EQUIPMENT DIVISION, JONES + BROTHER, SHEPLEY STREET, GUIDE BRIDGE, AUDENSHAW, MANCHESTER M34 5JD
TEL:061 330 6531 (10 LINES) 061 330 0111 (6 LINES) 061 330 3036 (4 LINES). TELEX: 669092. BROTHER INDUSTRIES LTD., NAGOYA, JAPAN.